

موارد تکمیلی پروژه های مهندسی نرم افزار - معماری نرم افزار پیشرفته برای طراحی نمودارها در DRAW.IO و blender کارشناسی ارشد

(پروژه های سیستم جامع مدیریت ریسک مدارس و دانشگاه)

تهیه کننده: دکتر مهران مختاری

نیم سال اول ۱۴۰۴

اینستاگرام @mehran.mokhtari.afra

کانال تلگرام t.me/mehran_mokhtari_afra

ایمیل mehrmokhtari@yahoo.com

ایمیل sama2dcad@gmail.com

ایمیل myclass1403@gmail.com

واتساپ https://wa.me/989119219626

ایتا https://eitaa.com/mehran1382369

پروژه ۱ - تحلیل ریسک نیازهای متغیر با مدل قیف تصمیم‌یار هوشمند

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (دانشجو، معلم، مدیر)	PK	user_id, username, role, department	User
نیازهای متغیر پروژه یا سیستم	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل نیازها	PK	model_id, name, version, description	FunnelModel
ثبت رویدادها و ریسک‌های تحلیل شده	PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسطه (User Requirement) M:N

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گم شده – کاربر ممکن است دسترسی به نیاز صحیح نداشته باشد	PK	INT	شناسه یکتا هر رکورد (User-Requirement)	ur_id
ارجاع نادرست – داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
ارجاع اشتباه – نیازهای واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
نقش اشتباه – تصمیمات تحلیل ممکن است اشتباه ثبت شود		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌های خاص جدول واسطه:

- مدیریت ارتباط M:N بین User و Requirement
- ریسک‌ها روی صحت داده‌ها و نقش‌ها تمرکز دارند
- در Draw.io سه بخش دارد: نام جدول / فیلدها / ریسک‌ها
- جزئیات DFD با رنگ و ضخامت خطوط برای ریسک نمایش داده می‌شوند

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

نوع خط	ضخامت خط	سطح هوشیاری	رنگ ریسک	نوع ریسک	نوع رابطه	نام رابطه	Entity هدف	Entity منبع	#
نقطه چین 2	متوسط	زرد	بالقوه	M:N	تخصیص به نیاز	User Requirement	1		
پر 3	بالا	قرمز	بالفعل	1:N	تحلیل توسط مدل	Requirement FunnelModel	2		
پر 3	بالا	قرمز	بالفعل	1:N	ثبت ریسک	User RiskLog	3		
نقطه چین 2	متوسط	زرد	بالقوه	1:N	مرتبط با نیاز	Requirement RiskLog	4		

نکات برای Draw.io

- هر موجودیت و جدول واسطه سه بخش دارد: نام / فیلدها / ریسک‌ها
- روابط بین موجودیت‌ها ضخامت و نوع خط بر اساس نوع ریسک نمایش داده می‌شود

- رنگ ریسک‌ها: سبز = کم، زرد = متوسط، قرمز = بالا
- خطوط: پر = بالفعل، نقطه‌چین = بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل سیستم با کاربران و محیط بیرونی به صورت کلی

جریان داده‌ها:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

موجودیت‌ها در DFD0:

نقش / توضیح	موجودیت
دانشجو، معلم یا مدیر – وارد کردن داده‌ها و دریافت گزارش	User
سیستم مرکزی تحلیل ریسک نیازها با مدل قیف تصمیم‌یار	Smart Funnel System
نمایش شاخص‌ها و گزارش‌های تحلیل نیاز	Dashboard

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی و جریان داده‌ها

فرآیندها:

۱. ثبت کاربران و نیازها
۲. مدیریت مدل تصمیم‌یار هوشمند
۳. تحلیل ریسک و ثبت رویدادها
۴. تولید گزارش و شاخص‌ها

جریان داده‌ها:

User → Process 1 → Requirement / User → Process 2 → FunnelModel → Process 3 → RiskLog → Process 4 → Dashboard → User

جدول فرآیندها و جریان داده:

خروجی	ورودی	Process
داده‌های اولیه User/Requirement و اطلاعات	1. ثبت کاربران و نیازها	Requirement و User

خروجی	ورودی	Process
مدل آماده تحلیل	اطلاعات FunnelModel	2. مدیریت مدل تصمیم‌یار
RiskLog / داده‌های ریسک	User, Requirement, FunnelModel	3. تحلیل ریسک
Dashboard / گزارش و شاخص‌ها RiskLog		4. تولید گزارش

DFD1: User, Requirement, FunnelModel, RiskLog, Dashboard در موجودیت‌ها

۴.۳ سطح دو (DFD2)

هدف: شکستن فرآیندها برای تحلیل دقیق

مثال – Process 4: تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها	Raw Data	KPI Report	User, RiskLog
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

DFD2: RiskLog, 4.1, 4.2, 4.3, Dashboard, User در موجودیت‌ها

جمع‌بندی: DFD

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها

۵. نمودار STD – State Transition Diagram

States: Idle, User_Registered, Requirement_Added, Analysis_Completed, Report_Displayed

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها

از وضعیت	به وضعیت	رویداد / شرط
Requirement_Added	Analysis_Completed	تحلیل توسط FunnelModel
Analysis_Completed	Report_Displayed	نمایش داشبورد
Report_Displayed	Idle	پایان چرخه

STD: User, System Admin, RiskLog, FunnelModel, States, Transitions

موجودیت‌های مرتبط

۶. نمودار Use Case

ارتباط	Use Case	Actor
Actor → System	ثبت اطلاعات	User
Actor → System	مشاهده Dashboard	User
Actor → System	مدیریت مدل Funnel	System Admin
Actor → System	مشاهده Dashboard	System Admin

موجودیت‌ها / ارتباط‌ها: User, System Admin, Dashboard, Use Cases مرتبط

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

جریان داده / هدف	نوع ارتباط	ارتباط با	Entity
ارسال اطلاعات و دریافت گزارش ارسال/دریافت			User Smart Funnel System
نمایش شاخص و گزارش	ارسال	Dashboard	Smart Funnel System
ارائه گزارش و شاخص‌ها	نمایش	User	Dashboard

DFD1:

جریان داده	ارتباط با	Entity
ارسال اطلاعات	Process 1	User
ذخیره داده‌ها	Process 1	User/Requirement
مدیریت مدل تحلیل	Process 2	FunnelModel

Entity	ارتباط با	جریان داده
Process 3	RiskLog	ثبت ریسک
Process 4	Dashboard	نمایش گزارش

DFD2:

Entity	ارتباط با	جریان داده
RiskLog	جمع‌آوری داده 4.1	
4.1	4.2 Raw Data	
4.2	4.3 KPI Report	
4.3 Dashboard	نمایش	

STD:

Entity	ارتباط با	نقش
User Transitions	عامل ایجاد و ثبت داده	
System Admin Transitions	عامل تغییر وضعیت و مدیریت	
RiskLog	States	ثبت وضعیت و اطلاعات

Use Case Diagram:

Actor	Use Case	جریان داده
User	ثبت داده	ارسال داده‌ها به سیستم
User Dashboard	مشاهده	دریافت گزارش‌ها
System Admin	Funnel مدل مدیریت داده‌ها و مدل	اصلاح و مدیریت داده‌ها
System Admin	Dashboard مشاهده	کنترل شاخص‌ها و گزارش‌ها

جمع‌بندی نهایی پروژه ۱ 

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها
- STD: وضعیت‌ها و رویدادها

- **Use Case:** تعامل Actor ها و سیستم
- **User Requirement:** جدول واسط دارای فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io

پروژه ۲ - پیش‌بینی تأخیر در تحویل پروژه نرم‌افزاری با الگوریتم یادگیری تقویتی

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (مدیر پروژه، توسعه‌دهنده، تستر)	INT, VARCHAR, VARCHAR, PK, VARCHAR	user_id, username, role, department	User
اطلاعات پروژه‌های نرم‌افزاری	INT, VARCHAR, DATE, DATE, PK, VARCHAR	project_id, name, start_date, end_date, status	Project
فعالیت‌ها و وظایف پروژه	INT, INT, VARCHAR, INT, PK, DATE, VARCHAR	task_id, project_id, name, assigned_to, deadline, status	Task
الگوریتم یادگیری تقویتی برای پیش‌بینی تأخیر	INT, VARCHAR, VARCHAR, PK, VARCHAR	model_id, name, version, parameters	RLModel
ثبت پیش‌بینی تأخیر وظایف پروژه	INT, INT, INT, VARCHAR, DATE, PK, VARCHAR	prediction_id, task_id, model_id, predicted_delay, date, notes	DelayPrediction

۲. جدول واسط (M:N (User_Project)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گم‌شده - دسترسی کاربران به پروژه و وظایف صحیح مختل می‌شود	PK	INT	شناسه یکتا هر رکورد- (User-Project)	up_id

رئسك (نام و توضيح)	كلىد	نوع داده	توضيح	ستون
ارجاع نادرست - داده‌های کاربر با پروژه واقعی مطابقت ندارد	INT FK → User(user_id)	ارجاع به جدول User	user_id	
ارجاع نادرست - وظایف و گزارش‌های پروژه دچار مشکل می‌شوند	FK → Project(project_id)	ارجاع به جدول Project	project_id	
نقش اشتباه - پیش‌بینی‌ها و تصمیمات نادرست ثبت می‌شود	VARCHAR	نقش کاربر در پروژه (Manager/Developer/Tester)	role	

ویژگی‌های خاص جدول واسط:

- مدیریت ارتباط M:N بین User و Project
- ریسک‌ها روی صحت داده‌ها و نقش‌ها تمرکز دارند
- در Draw.io سه بخش دارد: نام جدول / فیلدها / ریسک‌ها
- جزئیات DFD با رنگ و ضخامت خطوط برای ریسک نمایش داده می‌شوند

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Project	مشارکت در پروژه	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	Task	Project	وابسته به پروژه	1:N	بالفعل	قرمز	بالا	3	پر
3	Task DelayPrediction		پیش‌بینی تأخیر	1:N	بالفعل	قرمز	بالا	3	پر
4	RLModel DelayPrediction		تحلیل پیش‌بینی	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

نکات برای:  Draw.io

- هر موجودیت و جدول واسط سه بخش دارد: نام / فیلدها / ریسک‌ها
- روابط بین موجودیت‌ها ضخامت و نوع خط بر اساس نوع ریسک نمایش داده می‌شود
- رنگ ریسک‌ها: سبز = کم، زرد = متوسط، قرمز = بالا
- خطوط: پر = بالفعل، نقطه‌چین = بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل سیستم با کاربران و محیط بیرونی به صورت کلی

جریان داده‌ها:

[User] → [RL-Based Project Delay System] → [Dashboard / Reports] → [User]

موجودیت‌ها در: DFD0

موجودیت	نقش / توضیح
User	مدیر پروژه، توسعه‌دهنده یا تستر – وارد کردن اطلاعات پروژه و دریافت پیش‌بینی‌ها
RL-Based Project Delay System	سیستم مرکزی پیش‌بینی تأخیر با الگوریتم یادگیری تقویتی
Dashboard	نمایش شاخص‌ها، گزارش‌ها و پیش‌بینی تأخیر پروژه

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی و جریان داده‌ها

فرآیندها:

۱. ثبت کاربران و پروژه‌ها
۲. مدیریت وظایف پروژه
۳. اجرای مدل RL و پیش‌بینی تأخیر
۴. تولید گزارش و شاخص‌ها

جریان داده‌ها:

User → Process 1 → Project / User → Process 2 → Task → Process 3 → RLModel → Process 4 → DelayPrediction → Dashboard → User

جدول فرآیندها و جریان داده:

خروجی	ورودی	Process
داده‌های اولیه User/Project	اطلاعات User و Project	1. ثبت کاربران و پروژه‌ها
وظایف پروژه آماده تحلیل	اطلاعات Task	2. مدیریت وظایف پروژه
DelayPrediction / پیش‌بینی تأخیر	Task, RLModel	3. اجرای مدل RL
Dashboard / گزارش و شاخص‌ها	DelayPrediction	4. تولید گزارش

DFD1: User, Project, Task, RLModel, DelayPrediction, Dashboard

۴.۳ سطح دو (DFD2)

هدف: شکستن فرآیندها برای تحلیل دقیق

مثال – Process 4 – تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	DelayPrediction	Raw Data	DelayPrediction
4.2 محاسبه شاخص‌ها	Raw Data	KPI Report	User, DelayPrediction
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

DFD2: DelayPrediction, 4.1, 4.2, 4.3, Dashboard, User

جمع‌بندی: DFD

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها

۵. نمودار STD – State Transition Diagram

States: Idle, User_Registered, Task_Assigned, Prediction_Completed, Report_Displayed

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر

از وضعیت	به وضعیت	رویداد / شرط
User_Registered	Task_Assigned	ثبت وظایف پروژه
Task_Assigned	Prediction_Completed	اجرای مدل RL
Prediction_Completed	Report_Displayed	نمایش داشبورد
Report_Displayed	Idle	پایان چرخه

STD: User, System Admin, Task, RLModel, DelayPrediction, States, **موجودیت‌های مرتبط**
Transitions

۶. نمودار Use Case

Actor	Use Case	ارتباط
User	ثبت اطلاعات پروژه و وظایف	Actor → System
User	مشاهده داشبورد	Actor → System
System Admin	مدیریت مدل RL	Actor → System
System Admin	مشاهده داشبورد	Actor → System

موجودیت‌ها / ارتباط‌ها: User, System Admin, Dashboard, Use Cases مرتبط

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

جریان داده / هدف	نوع ارتباط	ارتباط با	Entity
ارسال اطلاعات پروژه و دریافت پیش‌بینی‌ها	ارسال/دریافت	RL-Based Project Delay System	User
نمایش شاخص و پیش‌بینی‌ها	ارسال	Dashboard	RL-Based Project Delay System
ارائه گزارش و شاخص‌ها	نمایش	User	Dashboard

DFD1:

Entity	ارتباط با	جریان داده
User	Process 1	ارسال اطلاعات
Process 1	User/Project	ذخیره داده‌ها
Process 2	Task	مدیریت وظایف
Process 3	RLModel	پیش‌بینی تأخیر
Process 4	Dashboard	نمایش گزارش

DFD2:

Entity	ارتباط با	جریان داده
DelayPrediction	4.1	جمع‌آوری داده
4.1	4.2	Raw Data
4.2	4.3	KPI Report
4.3	Dashboard	نمایش

STD:

Entity	ارتباط با	نقش
User Transitions		عامل ایجاد و ثبت داده
System Admin Transitions		عامل تغییر وضعیت و مدیریت
DelayPrediction	States	ثبت وضعیت و اطلاعات

Use Case Diagram:

Actor	Use Case	جریان داده
User	ثبت داده	ارسال داده‌ها به سیستم
User	مشاهده Dashboard	دریافت گزارش‌ها
System Admin	مدیریت مدل و داده‌ها	مدل RL
System Admin	کنترل شاخص‌ها و پیش‌بینی‌ها	مشاهده Dashboard

- **DFD0:** نمای کلی و جعبه سیاه
- **DFD1:** فرآیندهای اصلی + جریان داده‌ها
- **DFD2:** جزئیات دقیق فرآیندها
- **STD:** وضعیت‌ها و رویدادها
- **Use Case:** تعامل Actor ها و سیستم
- **User_Project:** جدول واسط دارای فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io

پروژه ۳ - مدیریت خطای ثبت‌نام کاربران در سامانه آموزشی با داشبورد پیش‌بینی خطا(3)

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (دانشجو، معلم، مدیر)	PK INT, VARCHAR, VARCHAR, VARCHAR	user_id, username, role, department	User
اطلاعات کلاس‌ها	PK INT, VARCHAR, VARCHAR	class_id, name, room	Class
تجهیزات شبکه (سوئیچ، روتر، کامپیوتر)	PK INT, VARCHAR, VARCHAR	device_id, type, location	Device
توپولوژی و شبکه سامانه آموزشی	PK INT, VARCHAR, VARCHAR	network_id, type, subnet	Network
ثبت رویدادهای ثبت‌نام و خطاهای کاربران	PK INT, INT, INT, VARCHAR, VARCHAR, DATE	log_id, user_id, class_id, event_type, severity, date	RegistrationLog

۲. جدول واسط M:N (User_Class)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
شناسه یکتا نادرست - رکورد تکراری یا گم‌شده باعث اختلال در عضویت کلاس و ثبت عملکرد دانشجو می‌شود	PK	INT	شناسه یکتا هر رکورد (User-Class)	uc_id

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
ارجاع نادرست به کاربر – داده‌های کاربر با کلاس واقعی همخوانی ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
ارجاع نادرست به کلاس – ثبت عملکرد و گزارش‌ها دچار مشکل می‌شود	FK → Class(class_id)	INT	ارجاع به جدول Class	class_id
نقش اشتباه – دسترسی و عملکرد کاربران اشتباه ثبت می‌شود	-	VARCHAR	نقش کاربر در کلاس (Student/Teacher)	role

ویژگی‌های خاص جدول واسط:

- مدیریت ارتباط M:N بین User و Class
- ریسک‌ها فقط روی صحت داده‌ها و کلیدها تمرکز دارند
- در Draw.io سه بخش دارد: نام جدول / فیلدها / ریسک‌ها
- جزئیات DFD مانند نوع خط، رنگ، سطح هوشیاری در نمودارها جداگانه نمایش داده می‌شوند

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Class	عضویت در کلاس	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	Device	Network	اتصال تجهیزات	1:N	بالفعل	قرمز	بالا	3	پر
3	User RegistrationLog		ثبت رویداد ثبت‌نام	1:N	بالفعل	قرمز	بالا	3	پر
4	Class RegistrationLog		مرتبط با کلاس	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

نکات برای Draw.io:

- هر موجودیت و جدول واسط سه بخش دارد: نام / فیلدها / ریسک‌ها
- روابط بین موجودیت‌ها ضخامت و نوع خط بر اساس نوع ریسک نمایش داده می‌شود
- رنگ ریسک‌ها: سبز = کم، زرد = متوسط، قرمز = بالا
- خطوط: پر = بالفعل، نقطه‌چین = بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل سیستم با کاربران و محیط بیرونی به صورت کلی

جریان داده‌ها:

[User] → [Registration Error Management System] → [Dashboard / Reports] → [User]

موجودیت‌ها در DFD0:

نقش / توضیح	موجودیت
دانشجو، معلم یا مدیر – ثبت اطلاعات و مشاهده پیش‌بینی خطاها	User
سیستم مرکزی مدیریت و پیش‌بینی خطاهای ثبت‌نام	Registration Error Management System
نمایش شاخص‌ها، گزارش‌ها و هشدارهای ثبت‌نام	Dashboard

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی و جریان داده‌ها

فرآیندها:

- ثبت کاربران و کلاس‌ها
- مدیریت تجهیزات و شبکه
- ثبت رویدادهای ثبت‌نام و خطاها
- تولید گزارش و شاخص‌ها

جریان داده‌ها:

User → Process 1 → Class / User → Process 2 → Device → Process 3 → RegistrationLog → Process 4 → Dashboard → User

جدول فرآیندها و جریان داده:

Process	ورودی	خروجی
1. ثبت کاربران و کلاس‌ها	اطلاعات User و Class	داده‌های اولیه User/Class
2. مدیریت تجهیزات و شبکه	اطلاعات Device	وضعیت شبکه

خروجی	ورودی	Process
داده‌های خطا و ثبت‌نام	RegistrationLog	3. ثبت رویدادهای ثبت‌نام
Dashboard / گزارش و شاخص‌ها	اطلاعات RegistrationLog	4. تولید گزارش و شاخص‌ها

موجودیت‌ها در DFD1: User, Class, Device, RegistrationLog, Dashboard

۴.۳ سطح دو (DFD2)

هدف: شکستن فرآیندها برای تحلیل دقیق

مثال – Process 4: تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RegistrationLog	Raw Data	RegistrationLog
4.2 محاسبه شاخص‌ها	Raw Data	KPI Report	User, RegistrationLog
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

موجودیت‌ها در DFD2: RegistrationLog, 4.1, 4.2, 4.3, Dashboard, User

جمع‌بندی: DFD

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها

۵. نمودار STD – State Transition Diagram

States: Idle, User_Registered, Data_Collected, KPI_Calculated, Report_Displayed

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Data_Collected	جمع‌آوری داده‌ها
Data_Collected	KPI_Calculated	محاسبه شاخص‌ها
KPI_Calculated	Report_Displayed	نمایش داشبورد

از وضعیت	به وضعیت	رویداد / شرط
Report_Displayed	Idle	پایان چرخه

STD: User, System Admin, RegistrationLog, Device/Network, States, موجودیت‌های مرتبط
Transitions

۶. نمودار Use Case

Actor	Use Case	ارتباط
User	Actor → System ثبت اطلاعات	
User	Dashboard مشاهده Actor → System	
System Admin	Actor → System مدیریت شبکه و تجهیزات	
System Admin	Dashboard مشاهده Actor → System	

موجودیت‌ها / ارتباط‌ها: User, System Admin, Dashboard, Use Cases مرتبط

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

جریان داده / هدف	نوع ارتباط	ارتباط با	Entity
ارسال اطلاعات و دریافت پیش‌بینی خطا	ارسال/دریافت	Registration Error Management System	User
نمایش شاخص و گزارش‌ها	ارسال	Dashboard	Registration Error Management System
ارائه گزارش و شاخص‌ها نمایش		User	Dashboard

DFD1:

جریان داده	ارتباط با	Entity
ارسال اطلاعات	Process 1	User
ذخیره داده‌ها	User/Class	Process 1
مدیریت تجهیزات و شبکه	Device	Process 2

Entity	ارتباط با	جریان داده
Process 3	RegistrationLog	ثبت رویدادهای ثبت نام
Process 4	Dashboard	نمایش گزارش و شاخص ها

DFD2:

Entity	ارتباط با	جریان داده
RegistrationLog	4.1	جمع آوری داده
4.1	4.2	Raw Data
4.2	4.3	KPI Report
4.3	Dashboard	نمایش

STD:

Entity	ارتباط با	نقش
User Transitions		عامل ایجاد و ثبت داده
System Admin Transitions		عامل تغییر وضعیت و مدیریت
RegistrationLog	States	ثبت وضعیت و اطلاعات

Use Case Diagram:

Actor	Use Case	جریان داده
User	ثبت داده	ارسال داده ها به سیستم
User	مشاهده Dashboard	دریافت گزارش ها
System Admin	مدیریت شبکه و تجهیزات	اصلاح و مدیریت
System Admin	مشاهده Dashboard	کنترل شاخص ها و پیش بینی ها

جمع بندی نهایی پروژه ۳ 

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده ها
- DFD2: جزئیات دقیق فرآیندها
- STD: وضعیت ها و رویدادها

- **Use Case:** تعامل Actor ها و سیستم
- **User_Class:** جدول واسط دارای فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io

پروژه ۴ - بهینه‌سازی تخصیص وظایف تیم توسعه با مدل داده‌محور و قیف عملکرد

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اعضای تیم توسعه (توسعه‌دهنده، مدیر پروژه، تستر)	INT, VARCHAR, VARCHAR, PK, VARCHAR	user_id, username, role, department	User
وظایف تیم توسعه	INT, VARCHAR, VARCHAR, PK, VARCHAR, VARCHAR	task_id, title, description, priority, status	Task
تیم‌های توسعه	INT, VARCHAR, INT, PK	team_id, name, lead_id	Team
ثبت تخصیص وظایف به کاربران	INT, INT, INT, DATE, PK, VARCHAR	log_id, task_id, user_id, date_assigned, status	AssignmentLog
شاخص‌های عملکرد برای وظایف	INT, INT, INT, PK, DECIMAL, DECIMAL	metric_id, task_id, user_id, efficiency, quality	PerformanceMetric

۲. جدول واسط (M:N (User_Task)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا حذف شده باعث تخصیص نادرست وظایف می‌شود	PK	INT	شناسه یکتا هر رکورد- (User-Task)	ut_id
ارجاع اشتباه به کاربر - وظایف به فرد نامناسب تخصیص داده می‌شود	FK → User(user_id)	INT	ارجاع به جدول User	user_id

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
ارجاع اشتباه به وظیفه – پیگیری و ارزیابی وظایف دچار خطا می‌شود	FK → Task(task_id)	INT	ارجاع به جدول Task	task_id
نقش اشتباه – عملکرد و شاخص‌ها اشتباه ثبت می‌شود	-	VARCHAR	نقش کاربر در وظیفه (Assignee/Reviewer)	role

ویژگی‌های خاص جدول واسط:

- مدیریت ارتباط M:N بین Task و User
- ریسک‌ها فقط روی صحت داده‌ها و کلیدها تمرکز دارند
- در Draw.io سه بخش دارد: نام جدول / فیلدها / ریسک‌ها
- جزئیات DFD مانند نوع خط، رنگ، سطح هوشیاری در نمودارها جداگانه نمایش داده می‌شوند

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Task	تخصیص وظایف	M:N	بالقوه	زرد	متوسط	نقطه‌چین 2
2	Task PerformanceMetric		ثبت عملکرد	1:N	بالفعل	قرمز	بالا	پر 3
3	Team	User	عضویت در تیم	1:N	بالقوه	زرد	متوسط	نقطه‌چین 2
4	AssignmentLog	Task	مرتبط با وظیفه	1:N	بالفعل	قرمز	بالا	پر 3

نکات برای Draw.io

- هر موجودیت و جدول واسط سه بخش دارد: نام / فیلدها / ریسک‌ها
- روابط بین موجودیت‌ها ضخامت و نوع خط بر اساس نوع ریسک نمایش داده می‌شود
- رنگ ریسک‌ها: سبز = کم، زرد = متوسط، قرمز = بالا
- خطوط: پر = بالفعل، نقطه‌چین = بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل سیستم با کاربران و سیستم‌های بیرونی به صورت کلی

جریان داده‌ها:

[User] → [Task Assignment & Performance System] → [Dashboard / Reports] → [User]

موجودیت‌ها در: DFD0

نقش / توضیح	موجودیت
اعضای تیم توسعه – دریافت و مشاهده وظایف و شاخص عملکرد	User
سیستم مرکزی مدیریت تخصیص وظایف و ارزیابی عملکرد	Task Assignment & Performance System
نمایش شاخص‌ها، گزارش‌ها و پیش‌بینی عملکرد کاربران	Dashboard

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی و جریان داده‌ها

فرآیندها:

- ثبت کاربران و تیم‌ها
- تعریف و مدیریت وظایف
- تخصیص وظایف و ثبت در AssignmentLog
- محاسبه شاخص‌های عملکرد و تولید گزارش

جریان داده‌ها:

User → Process 1 → Team/User → Process 2 → Task → Process 3 → AssignmentLog → Process 4 → PerformanceMetric → Dashboard → User

جدول فرآیندها و جریان داده:

Process	ورودی	خروجی
1. ثبت کاربران و تیم‌ها	اطلاعات User و Team	داده‌های اولیه User/Team
2. تعریف و مدیریت وظایف	اطلاعات Task	وظایف آماده تخصیص
3. تخصیص وظایف	اطلاعات AssignmentLog	ثبت تخصیص وظایف

خروجی	ورودی	Process
Dashboard / گزارش KPI و اطلاعات PerformanceMetric 4. محاسبه شاخص ها و گزارش		
موجودیت ها در DFD1: User, Team, Task, AssignmentLog, PerformanceMetric, Dashboard		

۴.۳ سطح دو (DFD2)

هدف: شکستن فرآیندها برای تحلیل دقیق

مثال – Process 4: محاسبه شاخص ها و گزارش

Sub-Process	Input	Output	Related Entities
4.1 جمع آوری داده ها	AssignmentLog	Raw Data	AssignmentLog
4.2 محاسبه شاخص ها	Raw Data	KPI Report Task, User, PerformanceMetric	
4.3 ارسال به داشبورد	KPI Report Dashboard Display		Dashboard, User

موجودیت ها در DFD2: AssignmentLog, 4.1, 4.2, 4.3, Dashboard, User

جمع بندی: DFD

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده ها
- DFD2: جزئیات دقیق فرآیندها

۵. نمودار STD – State Transition Diagram

States: Idle, Task_Assigned, Data_Collected, KPI_Calculated, Report_Displayed

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	Task_Assigned	تخصیص وظیفه
Task_Assigned	Data_Collected	جمع آوری داده ها
Data_Collected	KPI_Calculated	محاسبه شاخص ها
KPI_Calculated	Report_Displayed	نمایش داشبورد
Report_Displayed	Idle	پایان چرخه

STD: User, System Admin, Task, AssignmentLog, PerformanceMetric, مرتبط
States, Transitions

۶. نمودار Use Case

ارتباط	Use Case	Actor
Actor → System	مشاهده وظایف و ثبت وضعیت	User
Actor → System	مشاهده Dashboard و KPI	User
Actor → System	مدیریت وظایف و تیم	System Admin
Actor → System	مشاهده Dashboard و کنترل عملکرد	System Admin

موجودیت‌ها / ارتباط‌ها: User, System Admin, Task, Team, Dashboard

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

جریان داده / هدف	نوع ارتباط	ارتباط با	Entity
ارسال اطلاعات و دریافت وظایف و گزارش عملکرد	ارسال/دریافت	Task Assignment & Performance System	User
نمایش شاخص‌ها و گزارش	ارسال	Dashboard	Task Assignment & Performance System
ارائه گزارش KPI	نمایش	User	Dashboard

DFD1:

جریان داده	ارتباط با	Entity
ارسال اطلاعات	Process 1	User
ذخیره داده‌ها	User/Team	Process 1
مدیریت وظایف	Task	Process 2
ثبت تخصیص وظایف	AssignmentLog	Process 3
محاسبه KPI و نمایش در Dashboard	PerformanceMetric	Process 4

DFD2:

Entity	ارتباط با	جریان داده
AssignmentLog	4.1	جمع‌آوری داده‌ها
4.1	4.2	Raw Data
4.2	4.3	KPI Report
4.3 Dashboard		نمایش

STD:

Entity	ارتباط با	نقش
User Transitions		عامل ایجاد و ثبت داده
System Admin Transitions		مدیریت وظایف و کاربران
AssignmentLog	States	ثبت وضعیت و اطلاعات
PerformanceMetric	States	شاخص‌های عملکرد

Use Case Diagram:

Actor	Use Case	جریان داده
User	مشاهده وظایف و ثبت وضعیت	ارسال/دریافت داده‌ها
User	مشاهده KPI و Dashboard	دریافت گزارش‌ها
System Admin	مدیریت وظایف و تیم Task	اصلاح و مدیریت AssignmentLog
System Admin	مشاهده KPI و Dashboard	کنترل شاخص‌ها و عملکرد تیم

جمع‌بندی نهایی پروژه ۴ 

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها
- STD: وضعیت‌ها و رویدادها
- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Task دارای فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io

پروژه ۵ - کاهش ریسک خطاهای تست نرم افزار با مدل تصمیم‌یار مبتنی بر کیفیت کد

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اعضای تیم توسعه، تسترها و مدیر پروژه	INT, VARCHAR, VARCHAR, PK VARCHAR	user_id, username, role, department	User
ماژول‌های نرم افزار	INT, VARCHAR, INT, INT PK	module_id, name, complexity, lines_of_code	SoftwareModule
تست‌های نرم افزار مرتبط با ماژول‌ها	INT, INT, VARCHAR, PK VARCHAR, VARCHAR	test_id, module_id, description, priority, status	TestCase
ثبت تخصیص تست‌ها به کاربران و نتایج	INT, INT, INT, DATE, PK VARCHAR	log_id, test_id, user_id, date_assigned, result	AssignmentLog
شاخص‌های کیفیت کد برای هر ماژول	INT, INT, DECIMAL, PK DECIMAL, DECIMAL	metric_id, module_id, coverage, maintainability, defect_density	CodeQualityMetric

۲. جدول واسطه (M:N (User_TestCase)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا حذف شده باعث تخصیص نادرست تست‌ها می‌شود	PK	INT	شناسه یکتا هر رکورد (User- TestCase)	ut_id
ارجاع اشتباه به کاربر - تست به فرد نامناسب اختصاص داده می‌شود	FK → User(user_id)	INT	ارجاع به جدول User	user_id
ارجاع اشتباه به تست - داده‌ها و نتایج نامعتبر می‌شوند	FK → TestCase(test_id)	INT	ارجاع به جدول TestCase	test_id

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
نقش اشتباه – ثبت نتایج و ارزیابی عملکرد صحیح نخواهد بود		VARCHAR	نقش کاربر در تست (Tester/Reviewer)	role

✓ ویژگی‌های خاص جدول واسط:

- مدیریت ارتباط M:N بین User و TestCase
- ریسک‌ها فقط روی صحت داده‌ها و کلیدها تمرکز دارند
- در Draw.io سه بخش دارد: نام جدول / فیلدها / ریسک‌ها
- جزئیات DFD مانند نوع خط، رنگ، سطح هوشیاری در نمودارها جداگانه نمایش داده می‌شوند

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	رنگ ریسک	ضخامت سطح هوشیاری	نوع خط
1	User	TestCase	تخصیص تست	M:N	بالقوه	زرد	متوسط
2	TestCase	CodeQualityMetric	مرتبط با کیفیت کد	1:N	بالفعل	قرمز	بالا
3	SoftwareModule	TestCase	پوشش ماژول	1:N	بالفعل	قرمز	بالا
4	AssignmentLog	TestCase	ثبت عملکرد تست	1:N	بالفعل	قرمز	بالا

📌 نکات برای Draw.io

- هر موجودیت و جدول واسط سه بخش دارد: نام / فیلدها / ریسک‌ها
- روابط بین موجودیت‌ها ضخامت و نوع خط بر اساس نوع ریسک نمایش داده می‌شود
- رنگ ریسک‌ها: سبز = کم، زرد = متوسط، قرمز = بالا
- خطوط: پر = بالفعل، نقطه‌چین = بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0) 🌐

هدف: نمایش تعامل سیستم با کاربران و سیستم‌های تصمیم‌یار به صورت کلی

جریان داده‌ها:

[User] → [Test Management & Decision Support System] → [Dashboard / Reports] → [User]

موجودیت‌ها در DFD0:

نقش / توضیح	موجودیت
تستر، توسعه‌دهنده یا مدیر پروژه – وارد کردن داده‌های تست و مشاهده گزارش‌ها	User
سیستم مرکزی مدیریت تست و تحلیل کیفیت کد	Test Management & Decision Support System
نمایش شاخص‌ها و گزارش‌های مرتبط با تست و کیفیت کد	Dashboard

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی و جریان داده‌ها

فرآیندها:

- ثبت کاربران و ماژول‌ها
- تعریف و مدیریت تست‌ها
- تخصیص تست‌ها و ثبت نتایج در AssignmentLog
- محاسبه شاخص کیفیت کد و تولید گزارش

جریان داده‌ها:

User → Process 1 → SoftwareModule/User → Process 2 → TestCase → Process 3 → AssignmentLog → Process 4 → CodeQualityMetric → Dashboard → User

جدول فرآیندها و جریان داده:

خروجی	ورودی	Process
داده‌های اولیه User/SoftwareModule	اطلاعات User و SoftwareModule	1. ثبت کاربران و ماژول‌ها
تست‌ها آماده تخصیص	اطلاعات TestCase	2. تعریف و مدیریت تست‌ها
ثبت تخصیص و نتایج تست	اطلاعات AssignmentLog	3. تخصیص تست‌ها

خروجی	ورودی	Process
Dashboard / گزارش KPI	اطلاعات CodeQualityMetric	4. محاسبه شاخص‌ها و گزارش
DFD1: User, SoftwareModule, TestCase, AssignmentLog, CodeQualityMetric, Dashboard		

۴.۳ سطح دو (DFD2)

هدف: شکستن فرآیندها برای تحلیل دقیق

مثال – Process 4: محاسبه شاخص‌ها و گزارش

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	AssignmentLog	Raw Data	AssignmentLog
4.2 تحلیل کیفیت کد	Raw Data	Quality Report	SoftwareModule, CodeQualityMetric
4.3 ارسال به داشبورد	Quality Report	Dashboard Display	Dashboard, User

DFD2: AssignmentLog, 4.1, 4.2, 4.3, Dashboard, User

جمع‌بندی: DFD

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها

۵. نمودار STD – State Transition Diagram

States: Idle, Test_Assigned, Data_Collected, Quality_Analyzed, Report_Displayed

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	Test_Assigned	تخصیص تست
Test_Assigned	Data_Collected	اجرای تست و ثبت نتایج

رویداد / شرط	به وضعیت	از وضعیت
تحلیل کیفیت کد	Quality_Analyzed	Data_Collected
نمایش داشبورد و KPI	Report_Displayed	Quality_Analyzed
پایان چرخه	Idle	Report_Displayed

STD: User, System Admin, TestCase, AssignmentLog, موجودیت‌های مرتبط,
CodeQualityMetric, States, Transitions

۶. نمودار Use Case

ارتباط	Use Case	Actor
Actor → System	مشاهده تست‌ها و ثبت نتایج	User
Actor → System	مشاهده KPI و Dashboard	User
Actor → System	مدیریت تست‌ها و ماژول‌ها	System Admin
Actor → System	مشاهده Dashboard و تحلیل کیفیت کد	System Admin

موجودیت‌ها / ارتباط‌ها: User, System Admin, TestCase, SoftwareModule, Dashboard

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

جریان داده / هدف	نوع ارتباط	ارتباط با	Entity
ارسال اطلاعات و دریافت وظایف و گزارش کیفیت	ارسال/دریافت	Test Management & Decision Support System	User
نمایش شاخص‌ها و گزارش ارسال		Dashboard	Test Management & Decision Support System
ارائه گزارش و KPI	نمایش	User	Dashboard

DFD1:

جریان داده	ارتباط با	Entity
ارسال اطلاعات	Process 1	User

Entity	ارتباط با	جریان داده
Process 1 User/SoftwareModule		ذخیره داده‌ها
Process 2	TestCase	مدیریت تست‌ها
Process 3	AssignmentLog	ثبت تخصیص تست‌ها و نتایج
Process 4	CodeQualityMetric Dashboard	محاسبه KPI و نمایش در

DFD2:

Entity	ارتباط با	جریان داده
AssignmentLog	4.1	جمع‌آوری داده‌ها
4.1	4.2	Raw Data
4.2	4.3	Quality Report
4.3 Dashboard		نمایش

STD:

Entity	ارتباط با	نقش
User Transitions		عامل ایجاد و ثبت داده
System Admin Transitions		مدیریت تست‌ها و ماژول‌ها
AssignmentLog	States	ثبت وضعیت و اطلاعات
CodeQualityMetric	States	شاخص‌های کیفیت کد

Use Case Diagram:

Actor	Use Case	جریان داده
User	مشاهده تست‌ها و ثبت نتایج	ارسال/دریافت داده‌ها
User	مشاهده KPI و Dashboard	دریافت گزارش‌ها
System Admin	مدیریت تست‌ها و ماژول‌ها	اصلاح و AssignmentLog و TestCase
System Admin	مشاهده KPI و Dashboard	کنترل شاخص‌ها و عملکرد تست‌ها

- **DFD0:** نمای کلی و جعبه سیاه
- **DFD1:** فرآیندهای اصلی + جریان داده‌ها
- **DFD2:** جزئیات دقیق فرآیندها
- **STD:** وضعیت‌ها و رویدادها
- **Use Case:** تعامل Actor ها و سیستم
- **User_TestCase:** جدول واسطه دارای فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io

پروژه ۶ – تحلیل خودکار نیازها با Text Mining و ساخت مدل قیف تغییر نیاز

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اعضای تیم تحلیل نیازها، توسعه‌دهنده‌ها و مدیران پروژه	INT, VARCHAR, VARCHAR, PK VARCHAR	user_id, username, role, department	User
نیازهای نرم‌افزاری ثبت شده	INT, VARCHAR, VARCHAR, PK VARCHAR	req_id, title, description, priority, status	Requirement
منبع جمع‌آوری نیازها (مصاحبه، فرم، ایمیل)	INT, VARCHAR, PK VARCHAR, DATE	source_id, type, origin, date_collected	RequirementSource
خروجی تحلیل متن و استخراج شاخص‌های نیاز	INT, INT, VARCHAR, PK DECIMAL, VARCHAR	result_id, req_id, keywords, sentiment_score, category	TextMiningResult
مراحل مدل قیف تغییر نیاز	INT, VARCHAR, VARCHAR, PK DECIMAL	stage_id, name, description, probability	FunnelStage

۲. جدول واسطه (M:N (User_Requirement)

رِیسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا حذف شده باعث تخصیص نادرست تحلیل نیازها می‌شود	PK	INT	شناسه یکتا هر رکورد (User-Requirement)	ur_id
ارجاع اشتباه به کاربر – تحلیل نیازها به فرد نامناسب اختصاص داده می‌شود	FK → User(user_id)	INT	ارجاع به جدول User	user_id
ارجاع اشتباه به نیاز – تحلیل و گزارش‌ها نادرست خواهند بود	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
نقش اشتباه – ثبت نتایج تحلیل و اعتبارسنجی نادرست انجام می‌شود	-	VARCHAR	نقش کاربر در تحلیل نیاز (Analyst/Reviewer)	role

ویژگی‌های خاص جدول واسط:

- مدیریت ارتباط M:N بین User و Requirement
- ریسک‌ها فقط روی صحت داده‌ها و کلیدها تمرکز دارند
- در Draw.io سه بخش دارد: نام جدول / فیلدها / ریسک‌ها
- جزئیات DFD مانند نوع خط، رنگ، سطح هوشیاری در نمودارها جداگانه نمایش داده می‌شوند

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	ضخامت سطح هوشیاری	نوع خط
1	User	Requirement	تخصیص تحلیل	M:N	بالقوه	زرد	متوسط	نقطه چین 2
2	Requirement	TextMiningResult	تحلیل متن	1:N	بالفعل	قرمز	بالا	پر 3
3	RequirementSource	Requirement	منبع نیاز	1:N	بالفعل	قرمز	بالا	پر 3
4	FunnelStage	Requirement	قرارگیری در مرحله قیف	1:N	بالقوه	زرد	متوسط	نقطه چین 2

نکات برای Draw.io

- هر موجودیت و جدول واسط سه بخش دارد: نام / فیلدها / ریسک‌ها

- روابط بین موجودیت‌ها ضخامت و نوع خط بر اساس نوع ریسک نمایش داده می‌شود
- رنگ ریسک‌ها: سبز = کم، زرد = متوسط، قرمز = بالا
- خطوط: پر = بالفعل، نقطه‌چین = بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل سیستم با کاربران و سیستم‌های بیرونی به صورت کلی

جریان داده‌ها:

[User] → [Requirements Analysis System] → [Dashboard / Reports] → [User]

موجودیت‌ها در: DFD0

نقش / توضیح	موجودیت
تحلیل‌گر، توسعه‌دهنده یا مدیر پروژه – وارد کردن داده‌ها و مشاهده نتایج تحلیل	User
سیستم مرکزی تحلیل خودکار نیازها با Text Mining	Requirements Analysis System
نمایش شاخص‌ها و گزارش‌های نیازها و مدل قیف تغییر نیاز	Dashboard

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی و جریان داده‌ها

فرآیندها:

۱. ثبت کاربران و منابع نیازها
۲. جمع‌آوری و ثبت نیازها
۳. اجرای تحلیل متن (Text Mining)
۴. ساخت مدل قیف تغییر نیاز و تولید گزارش

جریان داده‌ها:

User → Process 1 → RequirementSource/User → Process 2 → Requirement → Process 3 → TextMiningResult → Process 4 → FunnelStage → Dashboard → User

جدول فرآیندها و جریان داده:

خروجی	ورودی	Process
داده‌های اولیه	اطلاعات User و RequirementSource	1. ثبت کاربران و منابع نیازها
نیازهای نرم‌افزاری آماده تحلیل	اطلاعات Requirement	2. جمع‌آوری و ثبت نیازها
TextMiningResult / شاخص‌ها و کلمات کلیدی	Requirement	3. اجرای تحلیل متن
FunnelStage / Dashboard / گزارش	TextMiningResult	4. ساخت مدل قیف تغییر نیاز

DFD1: User, Requirement, RequirementSource, TextMiningResult, FunnelStage, Dashboard

۴.۳ سطح دو (DFD2)

هدف: شکستن فرآیندها برای تحلیل دقیق

مثال – Process 4: ساخت مدل قیف تغییر نیاز

Sub-Process	Input	Output	Related Entities
4.1 دسته‌بندی نیازها	TextMiningResult	Categorized Requirements	TextMiningResult, Requirement
4.2 تعیین مرحله قیف	Categorized Requirements	Funnel Stage Assignment	FunnelStage, Requirement
4.3 ارسال به داشبورد	Funnel Stage Assignment	Dashboard Display	Dashboard, User

DFD2: TextMiningResult, 4.1, 4.2, 4.3, FunnelStage, Dashboard, User

جمع‌بندی: 

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها

States: Idle, User_Registered, Requirement_Collected, Text_Analyzed, Funnel_Assigned, Report_Displayed

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Collected	ثبت و جمع‌آوری نیازها
Requirement_Collected	Text_Analyzed	تحلیل متن و استخراج شاخص‌ها
Text_Analyzed	Funnel_Assigned	تعیین مرحله نیاز در قیف
Funnel_Assigned	Report_Displayed	نمایش داشبورد و KPI
Report_Displayed	Idle	پایان چرخه

STD: User, System Admin, Requirement, TextMiningResult, موجودیت‌های مرتبط, FunnelStage, Dashboard, States, Transitions

۶. نمودار Use Case

Actor	Use Case	ارتباط
User	ثبت اطلاعات نیازها	Actor → System
User	مشاهده Dashboard و تحلیل قیف	Actor → System
System Admin	مدیریت منابع نیازها و کاربران	Actor → System
System Admin	مشاهده Dashboard و KPI	Actor → System

موجودیت‌ها / ارتباط‌ها: User, System Admin, Requirement, FunnelStage, Dashboard

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

Entity	ارتباط با	نوع ارتباط	جریان داده / هدف
User	Requirements Analysis System	ارسال/دریافت	وارد کردن داده‌ها و دریافت تحلیل نیازها

Entity	ارتباط با	نوع ارتباط	جریان داده / هدف
Requirements Analysis System	Dashboard	ارسال	نمایش شاخص‌ها، مراحل قیف و گزارش‌ها
Dashboard	User	نمایش	ارائه گزارش‌ها و KPI

DFD1:

Entity	ارتباط با	جریان داده
User	Process 1	ارسال اطلاعات
Process 1	User/RequirementSource	ذخیره داده‌ها
Process 2	Requirement	ثبت نیازها
Process 3	TextMiningResult	تحلیل نیازها و استخراج شاخص‌ها
Process 4	FunnelStage Dashboard	تعیین مرحله قیف و نمایش در Dashboard

DFD2:

Entity	ارتباط با	جریان داده
TextMiningResult	4.1	دست‌بندی نیازها
4.1	4.2	تعیین مرحله قیف
4.2	4.3 Funnel Stage Assignment	
4.3 Dashboard		نمایش

STD:

Entity	ارتباط با	نقش
User Transitions		عامل ایجاد و ثبت داده
System Admin Transitions		مدیریت منابع نیازها و کاربران
Requirement	States	ثبت وضعیت نیازها
TextMiningResult/FunnelStage	States	تحلیل و تخصیص مرحله قیف

Use Case Diagram:

Actor	Use Case	جریان داده
User	ثبت نیازها	ارسال داده‌ها به سیستم
User	مشاهده Dashboard و قیف	دریافت گزارش‌ها و تحلیل‌ها
System Admin	مدیریت منابع نیازها	اصلاح و مدیریت Requirement و RequirementSource
System Admin	مشاهده Dashboard	کنترل شاخص‌ها و مراحل قیف

جمع‌بندی نهایی پروژه ۶

- **DFD0:** نمای کلی و جعبه سیاه
- **DFD1:** فرآیندهای اصلی + جریان داده‌ها
- **DFD2:** جزئیات دقیق فرآیندها
- **STD:** وضعیت‌ها و رویدادها
- **Use Case:** تعامل Actor ها و سیستم
- **User Requirement:** جدول واسط دارای فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io

پروژه ۷ – ارزیابی هوشمند مستندات پروژه با پرسشنامه کیفیت فرآیند (7) SDLC

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اعضای تیم پروژه، تحلیل‌گران و مدیران	INT, VARCHAR, VARCHAR, PK, VARCHAR	user_id, username, role, department	User
پروژه‌های نرم‌افزاری تحت ارزیابی	INT, VARCHAR, VARCHAR, DATE, PK, DATE	project_id, name, description, start_date, end_date	Project
مراحل SDLC مانند تحلیل، طراحی، توسعه، تست و استقرار	INT, VARCHAR, VARCHAR, PK, DECIMAL	process_id, name, description, weight	SDLCProcess

توضیح	Type Key	Attributes	Entity
پرسشنامه‌های مرتبط با کیفیت هر مرحله SDLC	INT, INT, VARCHAR, PK, VARCHAR	q_id, process_id, question_text, question_type	Questionnaire
نتایج ارزیابی پرسشنامه توسط کاربران	INT, INT, INT, INT, DECIMAL, PK, VARCHAR	eval_id, user_id, project_id, q_id, score, comments	EvaluationResult

۲. جدول واسط (M:N (User_Project)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا حذف شده باعث تخصیص نادرست ارزیابی می‌شود	PK	INT	شناسه یکتا هر رکورد (User-Project)	up_id
ارجاع اشتباه – امتیازدهی به پروژه اشتباه	FK → User(user_id)	INT	ارجاع به جدول User	user_id
ارجاع اشتباه – ارزیابی مستندات پروژه نادرست ثبت می‌شود	FK → Project(project_id)	INT	ارجاع به جدول Project	project_id
نقش اشتباه – ثبت نتایج و دسترسی کاربران نادرست خواهد بود	-	VARCHAR	نقش کاربر در ارزیابی (Evaluator/Reviewer)	role

✓ ویژگی‌های خاص جدول واسط:

- مدیریت ارتباط M:N بین User و Project
- ریسک‌ها فقط روی صحت داده‌ها و کلیدها تمرکز دارند
- در Draw.io سه بخش دارد: نام جدول / فیلدها / ریسک‌ها
- جزئیات DFD مانند نوع خط، رنگ، سطح هوشیاری در نمودارها جداگانه نمایش داده می‌شوند

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Project	تخصیص ارزیاب	M:N	بالقوه	زرد	متوسط	2	نقطه چین
2	Project	SDLCProcess	شامل مراحل SDLC	1:N	بالفعل	قرمز	بالا	3	پر
3	Questionnaire	SDLCProcess	مرتبط با مرحله	1:N	بالفعل	قرمز	بالا	3	پر
4	EvaluationResult	Questionnaire	ثبت پاسخ	1:N	بالقوه	زرد	متوسط	2	نقطه چین

📌 نکات برای Draw.io

- هر موجودیت و جدول واسط سه بخش دارد: نام / فیلدها / ریسکها
- روابط بین موجودیتها ضخامت و نوع خط بر اساس نوع ریسک نمایش داده می شود
- رنگ ریسکها: سبز = کم، زرد = متوسط، قرمز = بالا
- خطوط: پر = بالفعل، نقطه چین = بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل سیستم با کاربران و سیستمهای بیرونی به صورت کلی

جریان دادهها:

[User] → [SDLC Evaluation System] → [Dashboard / Reports] → [User]

موجودیتها در DFD0:

نقش / توضیح موجودیت

تحلیلگر، ارزیاب یا مدیر پروژه – وارد کردن دادهها و مشاهده نتایج ارزیابی User

SDLC Evaluation System

سیستم مرکزی ارزیابی هوشمند مستندات پروژه

Dashboard

نمایش شاخصها و گزارشهای کیفیت فرآیند SDLC

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی و جریان دادهها

فرآیندها:

1. ثبت کاربران و پروژه‌ها
2. ثبت پرسشنامه‌ها و ارتباط با SDLCProcess
3. جمع‌آوری نتایج ارزیابی (EvaluationResult)
4. تولید گزارش و شاخص‌های کیفیت

جریان داده‌ها:

User → Process 1 → Project/User → Process 2 → SDLCProcess → Process 3 →
Questionnaire → Process 3 → EvaluationResult → Process 4 → Dashboard → User

جدول فرآیندها و جریان داده:

Process	ورودی	خروجی
1. ثبت کاربران و پروژه‌ها	اطلاعات User و Project	داده‌های اولیه User/Project
2. ثبت پرسشنامه‌ها	اطلاعات SDLCProcess و Questionnaire	پرسشنامه‌های آماده برای ارزیابی
3. جمع‌آوری نتایج ارزیابی	EvaluationResult	داده‌های ارزیابی شده
4. تولید گزارش و شاخص‌ها	EvaluationResult	Dashboard / گزارش کیفیت SDLC

DFD1: User, Project, SDLCProcess, Questionnaire, EvaluationResult, Dashboard

۴.۳ سطح دو (DFD2)

هدف: شکستن فرآیندها برای تحلیل دقیق

مثال – Process 4: تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	EvaluationResult	Raw Data	EvaluationResult, Questionnaire
4.2 محاسبه شاخص‌ها	Raw Data	KPI Report	User, Project, SDLCProcess
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

DFD2: EvaluationResult, 4.1, 4.2, 4.3, Dashboard, User

جمع‌بندی: DFD

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها

۵. نمودار STD – State Transition Diagram

States: Idle, User_Registered, Questionnaire_Assigned, Data_Collected, KPI_Calculated, Report_Displayed

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Questionnaire_Assigned	تخصیص پرسشنامه به ارزیاب
Questionnaire_Assigned	Data_Collected	ثبت پاسخ‌ها و ارزیابی
Data_Collected	KPI_Calculated	محاسبه شاخص‌های کیفیت SDLC
KPI_Calculated	Report_Displayed	نمایش داشبورد و گزارش
Report_Displayed	Idle	پایان چرخه

STD: User, System Admin, Project, SDLCProcess, Questionnaire, موجودیت‌های مرتبط
EvaluationResult, Dashboard, States, Transitions

۶. نمودار Use Case

Actor	Use Case	ارتباط
User	ثبت اطلاعات پرسشنامه	Actor → System
User	مشاهده داشبورد	Actor → System
System Admin	مدیریت پروژه‌ها و پرسشنامه‌ها	Actor → System
System Admin	مشاهده داشبورد و KPI	Actor → System

: موجودیت‌ها / ارتباط‌ها, User, System Admin, Project, SDLCProcess, Questionnaire, Dashboard

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

جریان داده / هدف	نوع ارتباط	ارتباط با	Entity
وارد کردن داده‌ها و دریافت تحلیل مستندات	ارسال/دریافت	SDLC Evaluation System	User
نمایش شاخص‌ها و گزارش‌ها	ارسال	Dashboard	SDLC Evaluation System
ارائه گزارش‌ها و KPI	نمایش	User	Dashboard

DFD1:

جریان داده	ارتباط با	Entity
ارسال اطلاعات	Process 1	User
ذخیره داده‌ها	User/Project	Process 1
ثبت پرسشنامه‌ها	SDLCProcess/Questionnaire	Process 2
جمع‌آوری نتایج ارزیابی	EvaluationResult	Process 3
نمایش گزارش‌ها	Dashboard	Process 4

DFD2:

جریان داده	ارتباط با	Entity
جمع‌آوری داده‌ها 4.1		EvaluationResult
Raw Data	4.2	4.1
KPI Report	4.3	4.2
نمایش		4.3 Dashboard

STD:

نقش	ارتباط با	Entity
عامل ایجاد و ثبت داده		User Transitions
مدیریت منابع پروژه و پرسشنامه‌ها		System Admin Transitions

نقش	ارتباط با	Entity
ثبت وضعیت ارزیابی	States	EvaluationResult
تحلیل و تخصیص پرسشنامه	States	Questionnaire/SDLCProcess

Use Case Diagram:

جریان داده	Use Case	Actor
ارسال داده‌ها به سیستم	ثبت داده‌ها	User
دریافت گزارش‌ها و تحلیل‌ها	مشاهده User Dashboard	
اصلاح و مدیریت Project و Questionnaire	مدیریت پروژه‌ها	System Admin
کنترل شاخص‌ها و KPI	مشاهده System Admin Dashboard	

✓ جمع‌بندی نهایی پروژه ۷

- **DFD0:** نمای کلی و جعبه سیاه
- **DFD1:** فرآیندهای اصلی + جریان داده‌ها
- **DFD2:** جزئیات دقیق فرآیندها
- **STD:** وضعیت‌ها و رویدادها
- **Use Case:** تعامل Actor ها و سیستم
- **User_Project:** جدول واسط دارای فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io

پروژه ۸ - طراحی سیستم هشدار خطای تراکنش در اپلیکیشن‌های آنلاین 🚀

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
کاربران اپلیکیشن (مشتری، پشتیبان، مدیر)	PK	user_id, username, email, role	User
تراکنش‌های مالی ثبت‌شده توسط کاربران	PK	trans_id, user_id, amount, status, timestamp	Transaction

توضیح	Type Key	Attributes	Entity
مدل هوشمند تشخیص و هشدار خطای تراکنش	PK INT, VARCHAR, DECIMAL, VARCHAR	model_id, name, threshold, version	AlertModel
ثبت خطاها و هشدارهای تولید شده توسط سیستم	PK INT, INT, INT, VARCHAR, VARCHAR, DATE	log_id, trans_id, model_id, severity, message, date	ErrorLog

۲. جدول واسط (M:N (User_Transaction)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا حذف شده – تراکنش ممکن است دو بار پردازش شود	PK	INT	شناسه یکتا رکورد	ut_id
ارجاع اشتباه – کاربر نادرست به تراکنش متصل شده است	FK → User(user_id)	INT	ارجاع به جدول User	user_id
تطابق ناقص – تراکنش واقعی شناسایی نشده است	FK → Transaction(trans_id)	INT	ارجاع به جدول Transaction	trans_id
نقش اشتباه – خطا ممکن است به صورت نادرست تفسیر شود	-	VARCHAR	نقش کاربر (Customer/Support)	role

✓ ویژگی‌های خاص جدول واسط:

- مدیریت ارتباط M:N بین کاربران و تراکنش‌ها
- تمرکز ریسک بر صحت ارجاع و نوع نقش کاربر
- در Draw.io شامل سه بخش: نام جدول / فیلدها / ریسک‌ها
- نمایش ریسک‌ها با رنگ و ضخامت خطوط در DFD

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

نوع خط	ضخامت خط	سطح هوشیاری	رنگ ریسک	نوع ریسک	نوع رابطه	نام رابطه	Entity هدف	Entity منبع	#
نقطه چین 2	متوسط	زرد	بالمقوه	M:N	انجام تراکنش	User Transaction			1
پر 3	بالا	قرمز	بالفعل	1:N	تحلیل تراکنش‌ها	AlertModel	Transaction		2

نوع خط	ضخامت خط	سطح هوشیاری	رنگ ریسک	نوع ریسک	نوع رابطه	نام رابطه	Entity هدف	Entity منبع	#
پر	3	بالا	قرمز	بالفعل	1:N	ثبت هشدار	ErrorLog	AlertModel	3
نقطه‌چین	2	متوسط	زرد	بالقوه	1:N	منبع خطا	ErrorLog	Transaction	4

نکات برای Draw.io

- هر موجودیت سه بخش دارد: نام / فیلدها / ریسک‌ها
- رنگ‌ها: سبز = کم، زرد = متوسط، قرمز = بالا
- خطوط پر = ریسک بالفعل، نقطه‌چین = بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل سیستم هشدار با کاربران و محیط بیرونی

جریان داده‌ها:

[User] → [Smart Alert System] → [Dashboard / Notifications] → [User]

نقش / توضیح

موجودیت

User کاربر اپلیکیشن یا مدیر پشتیبانی

Smart Alert System سیستم هوشمند تشخیص خطای تراکنش

Dashboard نمایش هشدارها و وضعیت تراکنش‌ها

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی سیستم

فرآیندها:

۱. ثبت کاربران و تراکنش‌ها

۲. بارگذاری مدل هشدار هوشمند

۳. تحلیل تراکنش‌ها و تولید هشدار

۴. ثبت و نمایش خطاها

جریان داده‌ها:

User → P1 → Transaction / User → P2 → AlertModel → P3 → ErrorLog → P4 → Dashboard → User

Process	ورودی	خروجی
1 Transaction و User	اطلاعات	داده‌های اولیه ثبت‌شده
2 AlertModel	اطلاعات	مدل آماده برای تحلیل
3 Transaction, AlertModel		ErrorLog
4	ErrorLog	Dashboard / اعلان هشدار

موجودیت‌ها در DFD1: User, Transaction, AlertModel, ErrorLog, Dashboard

۴.۳ سطح دو (DFD2)

هدف: تحلیل جزئی فرآیند "تحلیل تراکنش‌ها و تولید هشدار"

Sub-Process	Input	Output	Related Entities
3.1 جمع‌آوری تراکنش‌ها	Transaction	Raw Data	Transaction
3.2 پردازش مدل هشدار	Raw Data	Error Candidates	AlertModel
3.3 ثبت هشدارها	Error Candidates	ErrorLog	ErrorLog, User

جمع‌بندی: DFD ✓

→ DFD0 • نمای کلی

→ DFD1 • فرآیندهای اصلی

→ DFD2 • جزئیات زیرفرآیند هشدار

۵. نمودار STD – State Transition Diagram ●

از وضعیت	به وضعیت	رویداد / شرط
Idle	Transaction_Received	ثبت تراکنش جدید
Transaction_Received	Analysis_Started	شروع تحلیل مدل هشدار
Analysis_Started	Alert_Generated	خطا شناسایی شد
Alert_Generated	Alert_Displayed	نمایش هشدار به کاربر
Alert_Displayed	Idle	بازگشت به حالت اولیه

موجودیت‌های مرتبط: User, AlertModel, ErrorLog, States, Transitions

۶. نمودار Use Case

ارتباط	Use Case	Actor
Actor → System	ثبت تراکنش	User
Actor → System	مشاهده هشدارها	User
Actor → System	مدیریت مدل هشدار	System Admin
Actor → System	مشاهده گزارشها	System Admin

۷. ارتباط موجودیتها و جریان دادهها

DFD0:

هدف	نوع ارتباط	ارتباط با	Entity
ثبت تراکنش / دریافت هشدار ارسال/دریافت			User Smart Alert System
نمایش هشدارها	ارسال	Dashboard	Smart Alert System
ارائه وضعیت تراکنشها	نمایش	User	Dashboard

DFD1:

جریان داده	ارتباط با	Entity
ثبت اطلاعات	Process 1	User
ذخیره دادهها	Transaction	Process 1
بارگذاری مدل	AlertModel	Process 2
ثبت هشدار	ErrorLog	Process 3
نمایش اعلانها	Dashboard	Process 4

DFD2:

جریان داده	ارتباط با	Entity
جمع آوری داده	3.1	Transaction
Raw Data	3.2	3.1
3.3 Error Candidates	3.2	3.2

جریان داده	ارتباط با	Entity
هشدارها		3.3 Dashboard

STD:

نقش	ارتباط با	Entity
ایجاد تراکنش		User Transitions
مدیریت مدل		System Admin Transitions
ذخیره هشدارها	States	ErrorLog

Use Case Diagram:

جریان داده	Use Case	Actor
ارسال داده‌ها	ثبت تراکنش	User
دریافت اعلان	مشاهده هشدار	User
اصلاح تنظیمات	مدیریت مدل هشدار	System Admin
بررسی وضعیت سیستم مشاهده گزارش‌ها		System Admin

✓ جمع‌بندی نهایی پروژه ۸

DFD0: تعامل کلی سیستم با کاربر

DFD1: فرآیندهای اصلی تحلیل تراکنش و هشدار

DFD2: جزئیات تحلیل خطا

STD: چرخه وضعیت تراکنش تا هشدار

Use Case: تعامل کاربر و مدیر با سیستم

User_Transaction: کنترل صحت داده و نقش کاربران

📌 پروژه ۹ – مدل‌سازی ریسک در چرخه عمر نرم‌افزار با قیف ۴ سطحی و داده تاریخی

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات پروژه‌های نرم‌افزاری در حال تحلیل	INT, VARCHAR, PK VARCHAR,	project_id, name, phase, manager, start_date, end_date	Project

توضیح	Type Key	Attributes	Entity
	VARCHAR, DATE, DATE		
فاکتورهای ریسک در مراحل مختلف چرخه عمر	PK	INT, VARCHAR, VARCHAR, VARCHAR, FLOAT, FLOAT	RiskFactor
مراحل قیف ۴ سطحی برای غربالسازی ریسک‌ها	PK	INT, VARCHAR, INT, VARCHAR	FunnelStage
داده‌های تاریخی از ریسک‌ها و پیامدهای آن‌ها	PK	INT, INT, VARCHAR, VARCHAR, DATE, VARCHAR	HistoricalData
نتایج مدل‌سازی ریسک و توصیه‌های هوشمند	PK	INT, INT, INT, INT, FLOAT, VARCHAR	RiskAnalysis

۲. جدول واسطه (RiskFactor_FunnelStage) M:N

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
داده تکراری یا گمشده – خطای نگاشت بین مرحله قیف و فاکتور ریسک	PK	INT	شناسه یکتا هر رکورد (Risk-FunnelStage)	rf_id
ارجاع نادرست – فاکتور ریسک در مرحله اشتباه ارزیابی شده	FK → RiskFactor(risk_id)	INT	ارجاع به جدول RiskFactor	risk_id
نگاشت اشتباه – سطح قیف نادرست تخصیص داده شده	FK → FunnelStage(stage_id)	INT	ارجاع به جدول FunnelStage	stage_id

ریسک (نام و توضیح) کلید	نوع داده	توضیح	ستون
احتمال عدم تطابق داده با نتیجه نهایی	-	وضعیت عبور از قیف status (Pending / Approved / VARCHAR Rejected)	
تأخیر یا ثبت ناقص زمان تحلیل	-	DATEIME زمان ثبت رکورد	timestamp

ویژگی‌های خاص جدول واسط: 

- پیوند بین مراحل قیف و فاکتورهای ریسک.
- قابل ردیابی بودن تصمیم‌ها در طول چرخه عمر.
- در **Draw.io** هر رکورد با رنگ مرحله قیف (آبی، سبز، نارنجی، قرمز) نمایش داده می‌شود.

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	Project	RiskFactor	شناسایی ریسک‌ها	1:N	بالقوه	زرد	متوسط	نقطه‌چین 2	
2	RiskFactor	FunnelStage	ارزیابی مرحله‌ای	M:N	بالفعل	قرمز	بالا	3	پر
3	FunnelStage	RiskAnalysis	تحلیل نهایی	1:N	بالفعل	قرمز	بالا	3	پر
4	RiskFactor	HistoricalData	بازخورد از داده تاریخی	1:N	بالقوه	نارنجی	متوسط	خط‌چین 2	
5	HistoricalData	RiskAnalysis	یادگیری از گذشته	1:N	هوشمند	سبز	بالا	3	پر

نکته: 

ضخامت و نوع خط در **Draw.io** با شدت ریسک هم‌راستا است.

قرمز = ریسک بالفعل / پررنگ،

زرد = بالقوه / نقطه‌چین،

سبز = هوشمند / پر خط ملایم.

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر  (DFD0)

هدف: نمایش تعامل بین کاربران، سیستم تحلیل ریسک، و داده تاریخی

جریان داده‌ها:

[User / Manager] → [Risk Funnel System] → [Dashboard / Reports] → [User / Manager]

نقش / توضیح	موجودیت
ورودی داده‌ها و تفسیر نتایج	User / Manager
مدل قیف ۴ سطحی تحلیل ریسک	Risk Funnel System
نمایش شاخص‌ها، روندها و گزارش‌ها	Dashboard

۴.۲ سطح یک (DFD1)

فرآیندها:

خروجی	ورودی	فرآیند
Dataset اولیه	اطلاعات پروژه + داده‌های تاریخی	1. جمع‌آوری داده‌ها
لیست فاکتورهای ریسک	Dataset	2. شناسایی ریسک‌ها
امتیازدهی و عبور مرحله‌ای فاکتورها		3. ارزیابی با قیف ۴ سطحی
گزارش تحلیلی و هشمنده	داده‌های قیف	4. تحلیل نهایی و توصیه
شاخص‌ها و گراف‌ها	نتایج تحلیل	5. تولید داشبورد

جریان داده:

User → P1 → P2 → P3 → P4 → P5 → Dashboard

۴.۳ سطح دو (DFD2)

مثال جزئی از Process 3 – ارزیابی با قیف ۴ سطحی

Sub-Process	Input	Output	Related Entities
3.1 فیلتر اولیه	RiskFactor	لیست فیلترشده سطح ۱	RiskFactor
3.2 ارزیابی احتمال	لیست سطح ۱	سطح ۲ با وزن‌دهی	FunnelStage
3.3 تحلیل تأثیر	داده سطح ۲	سطح ۳ امتیازدهی	RiskAnalysis
3.4 تأیید نهایی	داده سطح ۳	سطح ۴ (تصمیم)	FunnelStage, RiskAnalysis

۵. نمودار STD – State Transition Diagram

از وضعیت	به وضعیت	رویداد / شرط
Idle	Data_Collected	ورود داده‌ها
Data_Collected	Risks_Identified	شناسایی فاکتورهای ریسک
Risks_Identified	Funnel_Evaluation	آغاز تحلیل قیف
Funnel_Evaluation	Risk_Analyzed	نتیجه‌گیری از تحلیل
Risk_Analyzed	Report_Generated	تولید گزارش
Report_Generated	Idle	پایان چرخه

۶. نمودار Use Case

Actor	Use Case	ارتباط
Project Manager	ثبت پروژه	Actor → System
Analyst	شناسایی ریسک‌ها	Actor → System
System	اجرای قیف ۴ سطحی	System → Actor
System	تولید گزارش تحلیلی	System → Actor
Project Manager	مشاهده داشبورد	Actor ← System

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

Entity	ارتباط با	نوع ارتباط	جریان داده / هدف
User Risk Funnel System		دریافت	داده ورودی و خروجی تحلیلی ارسال
Risk Funnel System	Dashboard	ارسال	نمایش گزارش
Dashboard	User	بازخورد	بازنمایی نتایج

DFD1:

Entity	ارتباط با	جریان داده
User	Process 1	ارسال داده پروژه
Process 2	RiskFactor	استخراج فاکتورها
Process 3	FunnelStage	ارزیابی سطحی
Process 4	RiskAnalysis	نتایج تحلیلی
Process 5	Dashboard	خروجی نهایی

✓ جمع‌بندی نهایی پروژه ۹

بخش	توضیح
DFD0	نمای کلی چرخه مدل ریسک با داده تاریخی
DFD1	فرآیندهای کلیدی شناسایی تا تحلیل
DFD2	جزئیات قیف ۴ سطحی (ورودی، فیلتر، تحلیل، تصمیم)
STD	وضعیت‌ها از ورود داده تا گزارش نهایی
Use Case	نقش کاربران و تعامل با سیستم
Entity Table	ساختار پایگاه داده برای نگهداری داده‌های ریسک و قیف
	ارزیابی، بازخورد و یادگیری از داده تاریخی برای تصمیم‌گیری هوشمند روابط

🔗 پروژه ۱۰ - شناسایی گلوگاه‌های عملکردی با تحلیل داده‌های اجرای سیستم (Profiling Funnel)

۱. جدول پایه موجودیت‌ها و صفات (Entities)

Entity	Attributes	Type Key	توضیح
User	user_id, username, role, department	INT, VARCHAR, VARCHAR, VARCHAR PK	کاربران سیستم (توسعه‌دهنده، مدیر سرور، تحلیل‌گر عملکرد)

توضیح	Type Key	Attributes	Entity
ثبت داده‌های اجرای سیستم برای تحلیل عملکرد	INT, DATETIME, VARCHAR, FLOAT, PK FLOAT, FLOAT	exec_id, timestamp, process_name, cpu_usage, mem_usage, response_time	ExecutionLog
مدل قیف برای تحلیل و رتبه‌بندی گلوگاه‌ها	INT, VARCHAR, VARCHAR, PK VARCHAR	funnel_id, name, version, description	ProfilingFunnel
گزارش گلوگاه‌ها با شدت و محل وقوع	INT, INT, INT, VARCHAR, PK VARCHAR, VARCHAR, DATE	report_id, funnel_id, exec_id, severity, area, notes, date	BottleneckReport

۲. جدول واسطه (M:N (User_ExecutionLog

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
داده ناقص یا تکراری – تحلیل عملکرد نادرست انجام می‌شود	PK	INT	شناسه یکتا هر رکورد (User-ExecutionLog)	ue_id
ارجاع نادرست – کاربر با نقش اشتباه داده‌ها را وارد کرده است	FK → User(user_id)	INT	ارجاع به جدول User	user_id
داده‌های اجرایی نادرست – منجر به تشخیص غلط گلوگاه می‌شود	FK → ExecutionLog(exec_id)	INT	ارجاع به جدول ExecutionLog	exec_id
نقش اشتباه – داده‌های عملکردی نادرست تفسیر می‌شود	-	VARCHAR	نقش کاربر در تحلیل (Operator/Reviewer)	role

✓ ویژگی‌های خاص جدول واسطه:

- نمایش ارتباط بین کاربر و داده‌های اجرای سیستم
- کنترل صحت داده‌های پروفایلینگ
- تمرکز بر ریسک‌های ناشی از خطای انسانی در تحلیل داده‌ها

- در Draw.io سه بخش دارد: نام جدول / فیلدها / ریسک‌ها

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هشیاری	ضخامت خط	نوع خط
1	User	ExecutionLog	ثبت داده‌های اجرا	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	ExecutionLog	ProfilingFunnel	تحلیل عملکرد	1:N	بالفعل	قرمز	بالا	3	پر
3	ProfilingFunnel	BottleneckReport	تولید گزارش	1:N	بالفعل	قرمز	بالا	3	پر
4	User	BottleneckReport	مشاهده نتایج	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

نکات برای Draw.io

- رنگ خطوط بر اساس نوع ریسک
- ضخامت خطوط بر اساس سطح هشیاری
- سه بخش برای هر جدول: نام، فیلدها، ریسک‌ها
- پر (بالفعل) و نقطه‌چین (بالقوه)

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ - DFD0 - سطح صفر

هدف: نمایش کلی تعامل سیستم و کاربران

جریان داده‌ها:

[User] → [Profiling Funnel System] → [Performance Dashboard] → [User]

موجودیت

نقش / توضیح

واردکننده داده‌ها یا تحلیل‌گر عملکرد User

Profiling Funnel System سیستم مرکزی تحلیل گلوگاه‌ها

Performance Dashboard نمایش گزارش عملکرد و KPI ها

۴.۲ - DFD1 سطح یک

هدف: نمایش فرآیندهای اصلی سیستم تحلیل عملکرد

Process	ورودی	خروجی
1. ثبت کاربران و داده‌های اجرا	اطلاعات User و ExecutionLog	داده‌های اولیه ثبت شده
2. مدیریت مدل قیف تحلیل	ProfilingFunnel	مدل آماده تحلیل عملکرد اطلاعات
3. تحلیل گلوگاه‌ها	FunnelModel و ExecutionLog	BottleneckReport
4. تولید شاخص‌ها و گزارش‌ها	BottleneckReport	Dashboard / KPI

جریان داده‌ها:

User → Process 1 → ExecutionLog → Process 2 → ProfilingFunnel → Process 3 → BottleneckReport → Process 4 → Dashboard → User

۴.۳ - DFD2 سطح دو

نمونه: فرآیند ۳ - تحلیل گلوگاه‌ها

Sub-Process	Input	Output	Related Entities
3.1 جمع‌آوری داده‌های اجرا	ExecutionLog	Raw Performance Data	ExecutionLog
3.2 تحلیل قیفی داده‌ها	Raw Data	Bottleneck Areas	ProfilingFunnel
3.3 ثبت نتایج	Bottleneck Areas	BottleneckReport	BottleneckReport, User

نتیجه: ✓

- DFD0: دید کلان سیستم
- DFD1: جریان اصلی داده‌ها
- DFD2: فرآیندهای جزئی تحلیل

۵. نمودار STD - State Transition Diagram

وضعیت‌ها	توضیح
Idle	سیستم در حالت آماده‌به‌کار

وضعیت‌ها	توضیح
Data_Collected	داده‌های اجرا ثبت شده‌اند
Funnel_Applied	مدل قیف تحلیل اجرا شده است
Bottlenecks_Found	گلوگاه‌ها شناسایی شده‌اند
Report_Displayed	گزارش در داشبورد نمایش یافته است

از وضعیت	به وضعیت	رویداد / شرط
Idle	Data_Collected	ثبت داده‌های اجرا
Data_Collected	Funnel_Applied	شروع تحلیل قیفی
Funnel_Applied	Bottlenecks_Found	شناسایی گلوگاه‌ها
Bottlenecks_Found	Report_Displayed	تولید گزارش و نمایش
Report_Displayed	Idle	پایان چرخه تحلیل

۶. نمودار Use Case

Actor	Use Case	ارتباط
User	ثبت داده‌های اجرای سیستم	Actor → System
User	مشاهده داشبورد عملکرد	Actor ← System
System Admin	مدیریت مدل قیف تحلیل	Actor → System
System Admin	تأیید گزارش‌ها و مشاهده	Actor ← System

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

هدف	نوع ارتباط	ارتباط با	Entity
ارسال داده‌ها و مشاهده نتایج ارسال / دریافت			User Profiling Funnel System
نمایش شاخص‌ها	ارسال	Dashboard	Profiling Funnel System
ارائه گزارش‌ها	نمایش	User	Dashboard

DFD1:

Entity	ارتباط با	جریان داده
User	Process 1	ارسال داده‌ها
Process 1	ExecutionLog	ذخیره داده‌ها
Process 2	ProfilingFunnel	مدیریت مدل تحلیل
Process 3	BottleneckReport	ثبت نتایج
Process 4	Dashboard	نمایش گزارش‌ها

جمع‌بندی نهایی پروژه ۱۰

- **DFD0:** نمای کلی سیستم تحلیل عملکرد
- **DFD1:** فرآیندهای اصلی از ثبت داده تا گزارش‌دهی
- **DFD2:** تحلیل دقیق زیر فرآیندهای شناسایی گلوگاه
- **STD:** چرخه وضعیت از دریافت داده تا نمایش گزارش
- **Use Case:** تعامل کاربران و مدیران با سیستم
- **جدول واسط: User_ExecutionLog:** کنترل صحت و ریسک داده‌های عملکردی

پروژه ۱۱ – بهبود کیفیت کد با استفاده از شاخص‌های Maintainability و تصمیم‌یار ML

۱. جدول پایه موجودیت‌ها و صفات (Entities)

Entity	Attributes	Type Key	توضیح
Developer	dev_id, name, experience_level, team	INT, VARCHAR, VARCHAR, PK, VARCHAR	اطلاعات توسعه‌دهندگان پروژه
SourceCode	code_id, module_name, loc, cyclomatic_complexity, comment_ratio	INT, VARCHAR, INT, INT, FLOAT, PK	داده‌های ماژول‌های کد و شاخص‌های اولیه

توضیح	Type Key	Attributes	Entity
مدل تصمیم‌یار مبتنی بر یادگیری ماشین برای تحلیل کیفیت	PK INT, VARCHAR, VARCHAR, FLOAT, VARCHAR	model_id, name, version, accuracy, algorithm	MLModel
گزارش تحلیل کیفیت و سطح ریسک نگهداشت کد	PK INT, INT, INT, FLOAT, VARCHAR, DATE	report_id, code_id, model_id, maintainability_index, risk_level, date	QualityReport

۲. جدول واسطه (Developer_SourceCode) M:N

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا ناقص – کد بدون مالک مشخص	PK	INT	شناسه یکتا برای ارتباط توسعه‌دهنده و کد	dsc_id
ارجاع اشتباه – نسبت دادن کد به فرد اشتباه	FK → Developer(dev_id)	INT	ارجاع به جدول Developer	dev_id
ارتباط نادرست – کد واقعی با تحلیل ML تطبیق ندارد	FK → SourceCode(code_id)	INT	ارجاع به جدول SourceCode	code_id
نقش اشتباه – خطا در تعیین مسئولیت کیفیت	-	VARCHAR	نقش در توسعه (Author / Reviewer)	role

ویژگی‌های خاص:

- نگهداری ارتباط M:N بین توسعه‌دهنده و کد
- تمرکز بر ریسک‌های مربوط به مالکیت و بازبینی کد
- در Draw.io هر جدول سه بخش دارد: نام / فیلدها / ریسک‌ها
- ضخامت خطوط در روابط وابسته به شدت ریسک است

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

نوع خط	ضخامت خط	سطح هوشیاری	رنگ ریسک	نوع ریسک	نوع رابطه	نام رابطه	Entity هدف	Entity منبع	#
نقطه‌چین 2	متوسط	زرد	بالقوه	M:N	نگهداری و بازبینی		SourceCode	Developer	1

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
2	SourceCode	MLModel	تحلیل کیفیت با ML	1:N	بالفعل	قرمز	بالا	3	پر
3	MLModel	QualityReport	تولید گزارش کیفیت	1:N	بالفعل	قرمز	بالا	3	پر
4	SourceCode	QualityReport	وابستگی گزارش به مازول	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

📌 راهنمای طراحی در: Draw.io

- رنگ سبز = ریسک پایین، زرد = متوسط، قرمز = بالا
- خطوط پر برای ریسک بالفعل، نقطه‌چین برای بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل کلی سیستم تحلیل کیفیت کد با کاربران و محیط

جریان داده‌ها:

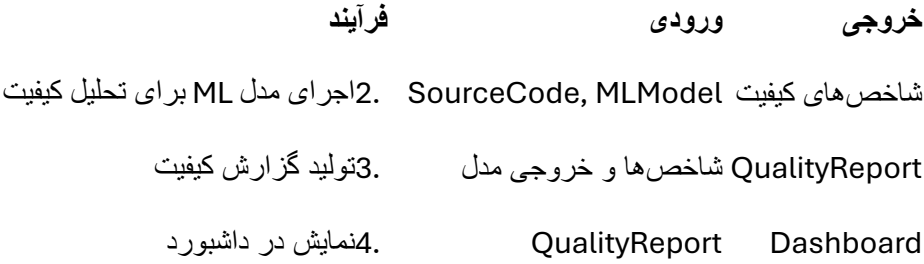
[Developer] → [ML-based Quality Analyzer System] → [Dashboard / Reports] → [Developer]

موجودیت	نقش / توضیح
Developer	تولید و بازبینی کد
ML-based Quality Analyzer	سیستم مرکزی تحلیل کیفیت
Dashboard Maintainability	نمایش شاخص‌های

۴.۲ سطح یک (DFD1)

فرآیندها:

خروجی	ورودی	فرآیند
داده‌های اولیه	Developer, SourceCode	1. ثبت توسعه‌دهندگان و کدها



جریان داده‌ها:

Developer → Process 1 → SourceCode → Process 2 → MLModel → Process 3 → QualityReport → Process 4 → Dashboard → Developer

۴.۳ سطح دو (DFD2)

مثال – Process 2: تحلیل کیفیت با مدل ML

Sub-Process	Input	Output	Related Entities
2.1 استخراج ویژگی‌های کد	SourceCode	Feature Vector	SourceCode
2.2 اجرای مدل ML	Feature Vector, MLModel	Prediction	MLModel
2.3 محاسبه شاخص نگهداشت	Prediction	Maintainability Index	QualityReport

جمع‌بندی DFD ها: ✓

- DFD0: نمای کلی سیستم
- DFD1: فرآیند اصلی تحلیل کیفیت
- DFD2: جزئیات تحلیل و تولید شاخص

۵. نمودار STD – State Transition Diagram

از وضعیت	به وضعیت	رویداد / شرط
Idle Developer_Registered		ثبت توسعه‌دهنده
Developer_Registered	Code_Uploaded	افزودن کد
Code_Uploaded	ML_Analyzed	اجرای مدل ML

رویداد / شرط به وضعیت از وضعیت

ML_Analyzed Report_Generated تولید گزارش کیفیت

Report_Generated Dashboard_Displayed نمایش شاخص‌ها

موجودیت‌های مرتبط: Developer, MLModel, QualityReport

۶. نمودار Use Case

ارتباط	Use Case	Actor
Actor → System	بارگذاری کد	Developer
Actor → System	مشاهده گزارش کیفیت	Developer
Actor → System	آموزش مدل ML	System Admin
Actor → System	به‌روزرسانی الگوریتم	System Admin

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

هدف	نوع ارتباط	ارتباط با	Entity
ارسال کد و دریافت گزارش ارسال/دریافت			Developer ML Analyzer
نمایش شاخص‌ها	ارسال		ML Analyzer Dashboard
ارائه نتایج	نمایش		Dashboard Developer

DFD1:

جریان داده	ارتباط با	Entity
ثبت داده	Process 1	Developer
تحلیل کیفیت	MLModel	Process 2
ثبت گزارش	QualityReport	Process 3
نمایش خروجی	Dashboard	Process 4

DFD2:

Entity	ارتباط با	جریان داده
SourceCode	2.1	استخراج ویژگی
2.1	2.2	Feature Vector
2.2	2.3	Maintainability Index
2.3	QualityReport	گزارش نهایی

جمع‌بندی نهایی پروژه ۱۱

- DFD0 دید کلی از تعامل توسعه‌دهنده و سیستم ML
- DFD1 فرآیندهای اصلی تحلیل کیفیت و تولید شاخص‌ها
- DFD2 مراحل دقیق پردازش ML و محاسبه Maintainability
- STD چرخه وضعیت از بارگذاری تا گزارش
- Use Case تعاملات کاربر و ادمین با سیستم
- جدول واسط → Developer_SourceCode نمایش ریسک مالکیت کد
- رنگ و ضخامت خطوط در Draw.io بیانگر شدت ریسک است

پروژه ۱۲ – شناسایی و کاهش خطای کاربران در فرم‌های آنلاین با مدل قیف تعامل کاربر (User Interaction Funnel)

۱. جدول پایه موجودیت‌ها و صفات (Entities)

Entity	Attributes	Type Key	توضیح
User	user_id, username, role, experience_level, device_type	INT, VARCHAR, VARCHAR, VARCHAR, PK	اطلاعات کاربران سیستم شامل نقش (دانشجو، کارمند)، سطح تجربه و نوع دستگاه
Form	form_id, name, type, version, num_fields	INT, VARCHAR, VARCHAR, PK, VARCHAR, INT	مشخصات فرم‌های آنلاین مورد بررسی

توضیح	Type Key	Attributes	Entity
ثبت کلیه تعاملات کاربر با فرم (ورود، کلیک، خطا)	INT, INT, INT, DATETIME, VARCHAR, BOOLEAN PK	log_id, user_id, form_id, timestamp, action_type, error_flag	InteractionLog
مراحل قیف تعامل کاربر (مشاهده، شروع، پر کردن، ارسال)	INT, VARCHAR, VARCHAR, FLOAT PK	stage_id, name, description, drop_rate	FunnelStage
الگوها و نوع خطاهای کاربران در هر مرحله از قیف تعامل	INT, INT, INT, VARCHAR, VARCHAR, VARCHAR PK	error_id, log_id, stage_id, error_type, severity, note	ErrorPattern

۲. جدول واسطه (User_Form) M:N

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – ممکن است کاربر چندبار در یک فرم خطا ثبت کند	PK	INT	شناسه یکتا هر رکورد (User-Form)	uf_id
ارجاع اشتباه – داده‌ها با کاربر واقعی منطبق نباشد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
فرم اشتباه – ارجاع به فرم نادرست باعث تحلیل غلط شود	FK → Form(form_id)	INT	ارجاع به جدول Form	form_id
مقدار اشتباه – وضعیت ناقص به‌صورت کامل ثبت شود	-	VARCHAR	وضعیت تکمیل فرم	completion_status
مقدار غیرواقعی – خطا در تحلیل کارایی کاربر	-	FLOAT	زمان صرف‌شده برای پر کردن فرم	time_spent

ویژگی‌های خاص: 

- مدیریت ارتباط M:N بین کاربران و فرم‌ها
- تمرکز بر تحلیل میزان خطا و تکمیل موفق
- ریسک‌ها مرتبط با داده‌های ناقص یا رفتار غیرواقعی کاربران

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	منبع Entity	هدف Entity	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Form	تعامل کاربر با فرم	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	Form	FunnelStage	عبور از مراحل قیف	1:N	بالفعل	قرمز	بالا	3	پر
3	InteractionLog	ErrorPattern	ثبت خطا	1:N	بالفعل	قرمز	بالا	3	پر
4	FunnelStage	ErrorPattern	تحلیل خطا در هر مرحله	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

نکات: Draw.io

- رنگ قرمز → خطای واقعی کاربر
- رنگ زرد → احتمال خطا در مراحل پرکردن
- خط پر → رخداد بالفعل
- خط نقطه‌چین → احتمال خطا

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل کلی کاربران با سیستم تحلیل قیف تعامل.

جریان داده‌ها:

[User] → [Form Interaction System] → [Dashboard / Error Report] → [User]

موجودیت

نقش / توضیح

User

پرکننده فرم یا کاربر نهایی

سیستم مرکزی جمع‌آوری و تحلیل تعاملات Form Interaction System

نمایش نتایج و نرخ خطا در مراحل قیف Dashboard

۴.۲ سطح یک (DFD1)

فرآیندها:

۱. ثبت تعاملات کاربران

۲. شناسایی مراحل قیف تعامل

۳. تحلیل خطا و زمان صرف شده

۴. تولید گزارش و شاخص های خطا

خروجی	ورودی	Process
داده های اولیه رفتار کاربر	InteractionLog	1. ثبت تعاملات کاربران
داده های مرحله بندی شده	InteractionLog, FunnelStage	2. شناسایی مراحل قیف
گزارش خطاها	داده های مرحله بندی شده ErrorPattern , 3. تحلیل خطا	
Dashboard	داده های خطا و زمان ها	4. تولید گزارش

۴.۳ سطح دو (DFD2)

Process 3 – تحلیل خطا

Sub-Process	Input	Output	Related Entities
3.1 تشخیص نوع خطا	InteractionLog	Error Type	InteractionLog
3.2 ارزیابی شدت خطا	Error Type	Severity Level	ErrorPattern
3.3 ذخیره در گزارش	Severity Level	Error Report	ErrorPattern, User

جمع بندی: DFD 

• DFD0: تعامل کلی کاربر و سیستم

• DFD1: فرآیندهای اصلی تحلیل

• DFD2: جزئیات تحلیل خطا

۵. نمودار وضعیت ها (STD – State Transition Diagram) 

از وضعیت	به وضعیت	رویداد / شرط
Idle	Form_Opened	کاربر فرم را باز می کند
Form_Opened	Form_Started	اولین فیلد پر می شود
Form_Started	Error_Detected	خطا در ورودی داده
Error_Detected	Correction_Made	اصلاح توسط کاربر

از وضعیت	به وضعیت	رویداد / شرط
Correction_Made	Form_Submitted	فرم ارسال می‌شود
Form_Submitted	Idle	چرخه کامل شد

موجودیت‌های مرتبط: User, Form, ErrorPattern, FunnelStage

۶. نمودار Use Case

Actor	Use Case	ارتباط
User	پر کردن فرم	Actor → System
User	مشاهده گزارش خطاهای خود	Actor → System
Admin	مدیریت مراحل قیف و ثبت فرم‌ها	Actor → System
Admin	مشاهده گزارش کلی خطاها	Actor → System

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

هدف	نوع ارتباط	ارتباط با	Entity
تعامل با فرم	ارسال داده‌ها	User Form Interaction	System
گزارش تعامل ارسال		Dashboard	Form Interaction System
ارائه بازخورد نمایش		User	Dashboard

DFD1:

جریان داده	ارتباط با	Entity
ثبت تعامل	Process 1	User
ذخیره رفتار	InteractionLog	Process 1
تعیین مرحله	FunnelStage	Process 2
تحلیل خطا	ErrorPattern	Process 3
نمایش گزارش	Dashboard	Process 4

DFD2:

Entity	جریان داده	ارتباط با
InteractionLog	3.1	داده خام
3.1	3.2	نوع خطا
3.2	3.3	شدت خطا
3.3 Dashboard	نمایش نهایی	

جمع‌بندی نهایی پروژه ۱۲

- تمرکز: کاهش خطای کاربران با تحلیل قیف تعامل (User Funnel Model)
- ابزار ترسیم Draw.io: برای DFD0 تا DFD2 ، Use Case و STD
- ویژگی‌ها:

- ثبت رفتار کاربر در مراحل مختلف فرم
- تشخیص محل و نوع خطا
- تحلیل شدت خطا و ارائه شاخص‌های عملکرد
- نمایش تعاملات و خطاها در داشبورد

نتیجه:

این مدل با بررسی مسیر تعامل کاربر از مشاهده تا ارسال فرم، خطاهای انسانی را شناسایی و با تحلیل قیفی آن‌ها، راهکارهایی برای کاهش نرخ رهاشدگی (Drop Rate) ارائه می‌دهد.

پروژه ۱۳ - پیش‌بینی میزان موفقیت پروژه بر اساس شاخص‌های ریسک تیم توسعه

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات اعضای تیم توسعه (برنامهنویس، تحلیلگر، طراح)	INT, VARCHAR, VARCHAR, PK, FLOAT	dev_id, name, role, experience_level, performance_score	Developer

توضیح	Type Key	Attributes	Entity
اطلاعات پروژه و شاخص‌های موفقیت آن	INT, VARCHAR, VARCHAR, INT, PK, FLOAT	proj_id, title, complexity, duration, success_score	Project
فهرست ریسک‌های مرتبط با عملکرد تیم و پروژه	INT, VARCHAR, VARCHAR, PK, VARCHAR, FLOAT	risk_id, name, type, impact_level, likelihood	RiskFactor
مدل ارزیابی برای پیش‌بینی موفقیت پروژه بر اساس ریسک تیم	INT, VARCHAR, VARCHAR, PK, VARCHAR, VARCHAR	model_id, name, version, method, description	EvaluationModel
ثبت ارزیابی ریسک‌های واقعی مشاهده‌شده در تیم توسعه	INT, INT, INT, INT, VARCHAR, DATE, PK, VARCHAR	log_id, dev_id, proj_id, risk_id, severity, date, notes	RiskLog

۲. جدول واسط (M:N (Developer_Project

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد ناقص یا تکراری – ارتباط نادرست بین عضو و پروژه	PK	INT	شناسه یکتا هر رکورد (Developer-Project)	dp_id
تخصیص اشتباه توسعه‌دهنده – تطابق مهارت با پروژه نادرست	FK → Developer(dev_id)	INT	ارجاع به جدول Developer	dev_id
ارتباط نادرست پروژه – ممکن است شاخص موفقیت اشتباه تحلیل شود	FK → Project(proj_id)	INT	ارجاع به جدول Project	proj_id
نقش اشتباه – اثر بر دقت پیش‌بینی موفقیت پروژه	-	VARCHAR	نقش عضو تیم در پروژه (Leader/Analyst/Tester)	role

ویژگی‌های خاص جدول واسط: 

- نمایش ارتباط چند به چند بین اعضای تیم و پروژه‌ها.
- فیلد نقش در ارزیابی عملکرد و ریسک‌ها اهمیت دارد.

- در **Draw.io** به صورت سه بخش نمایش داده می شود:
نام جدول / فیلدها / ریسک ها.
- رنگ و ضخامت خطوط برای نمایش نوع و شدت ریسک استفاده می شود.

۳. جدول روابط بین موجودیت ها با ویژگی های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	Developer	Project	مشارکت در پروژه	M:N	بالقوه	زرد	متوسط	2	نقطه چین
2	Project EvaluationModel		تحلیل موفقیت	1:N	بالفعل	قرمز	بالا	3	پر
3	Developer	RiskLog	ثبت ریسک عملکردی	1:N	بالفعل	قرمز	بالا	3	پر
4	RiskFactor	RiskLog	منبع ریسک	1:N	بالقوه	زرد	متوسط	2	نقطه چین
5	EvaluationModel	Project	پیش بینی نهایی	1:N	بالقوه	زرد	پایین	1	نقطه چین

📌 نکات طراحی برای Draw.io

- رنگ قرمز = ریسک بالا، زرد = متوسط، سبز = پایین
- خطوط پر = ریسک بالفعل، نقطه چین = بالقوه
- ضخامت خط = شدت ریسک (۱ کم، ۲ متوسط، ۳ بالا)

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

🎯 هدف: نمایش کلی جریان داده بین کاربران، سیستم و مدل ارزیابی

جریان داده ها:

[Developer / Project Manager] → [Smart Evaluation System] → [Dashboard / Reports] → [Project Manager]

موجودیت

نقش / توضیح

Developer

ثبت داده های عملکردی و شاخص های تیم

نقش / توضیح

موجودیت

Project Manager

مشاهده پیش‌بینی موفقیت و گزارش‌ها

سیستم مرکزی تحلیل ریسک و پیش‌بینی موفقیت Smart Evaluation System

نمایش درصد موفقیت پروژه و شاخص‌های کلیدی Dashboard

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی تحلیل ریسک تیم توسعه

Process	ورودی	خروجی
1. ثبت اعضا و پروژه‌ها	داده‌های Developer و Project	داده‌های اولیه ثبت‌شده
2. ثبت و تحلیل ریسک‌ها	Developer ، RiskFactor	RiskLog
3. اجرای مدل پیش‌بینی	EvaluationModel ، RiskLog	پیش‌بینی موفقیت پروژه
4. تولید داشبورد و گزارش	Dashboard و گزارش نهایی پیش‌بینی‌ها و شاخص‌ها	

جریان داده:

Developer → Process 1 → Project → Process 2 → RiskLog → Process 3 → EvaluationModel → Process 4 → Dashboard → Project Manager

۴.۳ سطح دو (DFD2)

مثال – Process 3: اجرای مدل پیش‌بینی

Sub-Process	Input	Output	Related Entities
3.1 جمع‌آوری داده‌های ریسک	RiskLog	Raw Risk Data	Developer, RiskLog
3.2 تحلیل شاخص‌ها	Raw Risk Data	Risk Index	EvaluationModel
3.3 پیش‌بینی موفقیت	Risk Index	Success Prediction	Project, Dashboard

نکات: ✓

- هر فرآیند با رنگ‌بندی ریسک نمایش داده شود.
- خروجی نهایی شامل درصد موفقیت پروژه و سطح ریسک کلی است.

۵. نمودار STD – State Transition Diagram

از وضعیت	به وضعیت	رویداد / شرط
Idle	Data_Collected	ورود داده‌های تیم و پروژه
Data_Collected	Risk_Analyzed	تحلیل شاخص‌های ریسک
Risk_Analyzed	Prediction_Generated	اجرای مدل ارزیابی
Prediction_Generated	Dashboard_Displayed	نمایش گزارش
Dashboard_Displayed	Idle	پایان چرخه ارزیابی

موجودیت‌های مرتبط: Developer, Project Manager, RiskLog, EvaluationModel

۶. نمودار Use Case

Actor	Use Case	ارتباط
Developer	ثبت عملکرد و شاخص‌ها	Actor → System
Project Manager	مشاهده گزارش موفقیت	Actor → System
System Admin	تنظیم مدل Evaluation	Actor → System
System Admin	مشاهده شاخص‌های کلی	Actor → System

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

هدف	نوع ارتباط	ارتباط با	Entity
ثبت داده و مشاهده نتایج ارسال/دریافت			Developer Evaluation System
نمایش موفقیت پروژه	ارسال	Dashboard	Evaluation System
گزارش نهایی پیش‌بینی نمایش		Project Manager	Dashboard

DFD1

جریان داده	ارتباط با	Entity
داده‌های عملکردی	Process 1	Developer

Entity	ارتباط با	جریان داده
Process 2	RiskLog	ثبت ریسک‌ها
Process 3	EvaluationModel	اجرای تحلیل
Process 4	Dashboard	ارسال گزارش نهایی

جمع‌بندی نهایی پروژه ۱۳

- DFD0: دید کلی از سیستم پیش‌بینی موفقیت پروژه
- DFD1: فرآیندهای کلیدی از جمع‌آوری تا پیش‌بینی
- DFD2: تحلیل دقیق و گام‌به‌گام اجرای مدل
- STD: وضعیت‌ها و انتقال‌ها در فرآیند ارزیابی
- Use Case: تعامل کاربران (توسعه‌دهنده، مدیر پروژه، مدیر سیستم)
- جدول واسط: Developer_Project مدیریت نقش‌ها و ریسک‌های انسانی در موفقیت پروژه
- تمرکز تحلیلی: ارزیابی کمی و کیفی ریسک‌ها در سطح تیم توسعه و مدل پیش‌بینی موفقیت

پروژه ۱۴ - ساخت مدل قیف بازخورد کاربر برای بهبود تجربه نرم‌افزار (UX Funnel)

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
کاربران نرم‌افزار (کاربر عادی، تست‌کننده، مدیر UX)	INT, VARCHAR, VARCHAR, PK, VARCHAR	user_id, username, role, device_type	User
بازخوردهای کاربران درباره تجربه استفاده	INT, INT, VARCHAR, INT, PK, VARCHAR, DATE	fb_id, user_id, category, satisfaction_level, comment, date	Feedback
مدل قیف UX برای تحلیل مسیر و بازخورد کاربر	INT, VARCHAR, VARCHAR, PK, VARCHAR	model_id, name, version, description	UXFunnelModel

توضیح	Type Key	Attributes	Entity
گزارش نهایی تحلیل UX و شاخص‌های تجربه کاربر	INT, INT, FLOAT, VARCHAR, PK VARCHAR, DATE	report_id, model_id, avg_satisfaction, critical_issues, summary, date	UXReport

۲. جدول واسط (User_Feedback) M:N

رِیسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد گمشده یا تکراری – ممکن است بازخورد کاربر دوباره شمارش شود	PK	INT	شناسه یکتا هر رکورد (User-Feedback)	uf_id
ناهماهنگی کاربر – شناسه اشتباه موجب خطا در تحلیل رفتار می‌شود	FK → User(user_id)	INT	ارجاع به جدول User	user_id
بازخورد اشتباه – داده به کاربر نادرست نسبت داده می‌شود	FK → Feedback(fb_id)	INT	ارجاع به جدول Feedback	fb_id
ناسازگاری بین پلتفرم‌ها (مثلاً موبایل/وب) می‌تواند UX را تحریف کند	–	VARCHAR	نوع دستگاه کاربر	device_type

✓ ویژگی‌های جدول واسط:

- ایجاد ارتباط بین کاربر و بازخوردها در قیف UX
- داده‌ها برای شناسایی مسیر تعامل کاربر در سیستم استفاده می‌شوند
- رِیسک‌ها روی تکرار داده، صحت کاربر و دستگاه تمرکز دارند
- در Draw.io هر جدول سه بخش دارد: نام، فیلدها، و رِیسک‌ها

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های رِیسک

نوع خط	ضخامت خط	رنگ رِیسک	نوع رِیسک	نوع رابطه	نام رابطه	Entity هدف	Entity منبع	#
نقطه‌چین 2	متوسط	زرد	بالقوه	1:N	ارسال بازخورد	Feedback	User	1

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
2	Feedback UXFunnelModel	UXFunnelModel	تحلیل بازخورد	1:N	بالفعل	قرمز	بالا	3	پر
3	UXFunnelModel	UXReport	تولید گزارش UX	1:N	بالفعل	قرمز	بالا	3	پر
4	User	UXReport	مشاهده شاخص‌ها	1:N	بالقوه	سبز	کم	1	نقطه‌چین

📌 راهنمای طراحی در Draw.io

- رنگ سبز = ریسک کم، زرد = متوسط، قرمز = بالا
- خط پر = بالفعل، نقطه‌چین = بالقوه
- ضخامت خطوط متناسب با سطح هوشیاری (۲ متوسط، ۳ بالا)

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل کلی کاربر با مدل قیف UX.

جریان داده‌ها:

[User] → [UX Funnel System] → [UX Report Dashboard] → [User]

موجودیت

نقش / توضیح

User

ارسال بازخورد، مشاهده گزارش UX

UX Funnel System

تحلیل داده‌های بازخورد و مسیر کاربر

UX Report Dashboard

نمایش شاخص‌های رضایت و تجربه کاربر

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی در تحلیل قیف UX.

Process

ورودی

خروجی

1. ثبت کاربران و بازخوردها

User, Feedback داده‌های

داده‌های اولیه قیف UX

خروجی	ورودی	Process
مسیرهای بهینه UX	Feedback, UXFunnelModel	2. تحلیل مسیر کاربر
RiskLog (درون سیستمی) UX	داده های مسیر	3. شناسایی نقاط ضعف UX
UXReport و شاخص های UX	داده های مدل و بازخورد	4. تولید گزارش UX

جریان کلی:

User → Process1 → Feedback → Process2 → UXFunnelModel → Process3 → Process4 → UXReport → User

۴.۳ سطح دو (DFD2)

مثال – Process 4: تولید گزارش UX

Sub-Process	Input	Output	Related Entities
4.1 جمع آوری داده های UX	Feedback	Raw UX Data	Feedback
4.2 تحلیل شاخص های مسیر	Raw Data	Metrics	UXFunnelModel
4.3 تولید داشبورد UX	Metrics	UXReport	UXReport, User

جمع بندی: DFD ✓

- DFD0: نمای کلی و چرخه بازخورد کاربر
- DFD1: فرآیندهای اصلی تحلیل UX
- DFD2: تحلیل جزئی داده و تولید داشبورد

۵. نمودار STD – State Transition Diagram

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Active	ورود کاربر
User_Active	Feedback_Submitted	ارسال بازخورد
Feedback_Submitted	Analyzed	تحلیل بازخورد در مدل UX
Analyzed	Report_Generated	تولید گزارش UX
Report_Generated	Idle	پایان چرخه UX

موجودیت‌های مرتبط : User, UXFunnelModel, UXReport, States, Transitions

۶. نمودار Use Case

ارتباط	Use Case	Actor
Actor → System	ارسال بازخورد	User
Actor → System	مشاهده نتایج UX	User
Actor → System	تحلیل مسیر کاربر	UX Analyst
Actor → System	مدیریت مدل قیف UX	System Admin

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

هدف	نوع ارتباط	ارتباط با	Entity
تعامل بازخورد و گزارش ارسالی/دریافت		UX Funnel System	User
تولید و نمایش نتایج UX	ارسال	UX Report Dashboard	UX Funnel System
ارائه شاخص‌های UX	نمایش	User	Dashboard

DFD1:

جریان داده	ارتباط با	Entity
ارسال بازخورد	Process1	User
ذخیره بازخورد	Feedback	Process1
تحلیل مسیر کاربر	UXFunnelModel	Process2
ثبت مشکلات UX	UX RiskLog	Process3
تولید شاخص و گزارش UXReport		Process4

جمع‌بندی نهایی پروژه ۱۴

- DFD0: چرخه بازخورد و تحلیل تجربه کاربر
- DFD1: فرآیندهای اصلی مدل قیف UX

- DFD2: تحلیل جزئی داده‌ها و شاخص‌های UX
- STD: تغییر وضعیت از ارسال بازخورد تا تولید گزارش
- Use Case: نقش کاربران، تحلیل‌گر UX، و مدیر سیستم
- جدول واسط: User_Feedback مدیریت ارتباط کاربران با بازخوردها
- سطح هوشمندی: مدل قیف UX با شاخص‌های دینامیک برای بهبود تجربه

🔗 پروژه ۱۵ - تحلیل ریسک امنیتی در مراحل توسعه با تصمیم‌گیری هوشمند و گزارش‌دهی خودکار

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
کاربران سامانه (تحلیل‌گر، مدیر پروژه، توسعه‌دهنده)	INT, VARCHAR, VARCHAR, PK, VARCHAR	user_id, username, role, department	User
مراحل امنیتی در چرخه توسعه نرم‌افزار	INT, VARCHAR, VARCHAR, INT, PK	phase_id, name, description, priority	SecurityPhase
مدل تصمیم‌یار هوشمند برای تحلیل ریسک‌ها	INT, VARCHAR, VARCHAR, PK, VARCHAR, VARCHAR	model_id, name, version, algorithm, description	AI_DecisionModel
رخداد‌های ریسک شناسایی‌شده در مراحل مختلف	INT, INT, INT, VARCHAR, VARCHAR, PK, VARCHAR, DATE, VARCHAR	risk_id, phase_id, model_id, severity, likelihood, detection, date, notes	RiskEvent
گزارش‌های خودکار تولیدشده توسط سیستم هوشمند	INT, INT, VARCHAR, PK, DATE, VARCHAR	report_id, risk_id, generated_by, date, summary	ReportLog

۲. جدول واسط (User_Phase) M:N

ریسک (نام و توضیح) کلید	نوع داده	توضیح	ستون
ریسک رکورد تکراری – تخصیص اشتباه مرحله به کاربر	INT	شناسه یکتا هر رکورد User-Phase	up_id
ریسک ارجاع نادرست – کاربر غیرمجاز به مرحله حساس دسترسی دارد	INT	ارجاع به جدول User	user_id
ریسک ارجاع اشتباه – مرحله امنیتی خارج از مسئولیت کاربر انتخاب می‌شود	INT	ارجاع به جدول SecurityPhase	phase_id
نقش اشتباه – تحلیل نادرست یا تضاد منافع – در ارزیابی ریسک	VARCHAR	نقش کاربر در فاز امنیتی role (Developer / Auditor / Manager)	

ویژگی‌های خاص جدول واسط: 

- ارتباط بین User و SecurityPhase به صورت M:N
- تمرکز بر ریسک تخصیص وظایف امنیتی
- برای نمایش در Draw.io شامل سه بخش است: نام جدول، فیلدها، و ریسک‌ها
- رنگ زرد برای ریسک‌های متوسط، قرمز برای ریسک‌های بالا استفاده شود

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	SecurityPhase	تخصیص نقش امنیتی	M:N	بالقوه	زرد	متوسط	2	نقطه چین
2	SecurityPhase	RiskEvent	ایجاد ریسک در فاز	1:N	بالفعل	قرمز	بالا	3	پر
3	AI_DecisionModel	RiskEvent	تحلیل و ارزیابی ریسک	1:N	بالفعل	قرمز	بالا	3	پر

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
4	RiskEvent	ReportLog	تولید گزارش خودکار	1:N	بالقوه	زرد	متوسط	نقطه چین 2	
5	User	ReportLog	مشاهده و تایید گزارش	1:N	بالقوه	سبز	کم	نازک 1	

نکات برای Draw.io 

- روابط قرمز = ریسک بالا
- خطوط پر = ریسک بالفعل
- نقطه چین = ریسک بالقوه
- رنگ سبز برای فعالیت های تأییدی و ایمن

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر  (DFD0)

هدف: نمایش کلی تعامل کاربران با سیستم تصمیمیار امنیتی

جریان داده ها:

[User] → [Smart Security Analyzer] → [Dashboard / Auto Reports] → [User]

موجودیت

نقش / توضیح

User کاربر انسانی (تحلیلگر، مدیر پروژه، توسعه دهنده)

Smart Security Analyzer سیستم مرکزی تصمیمیار هوشمند برای تحلیل ریسک های امنیتی

Dashboard نمایش نتایج، وضعیت مراحل امنیتی و گزارش های خودکار

۴.۲ سطح یک  (DFD1)

هدف: نمایش فرآیندهای اصلی و داده های در جریان

Process

ورودی

خروجی

داده های اولیه فازها و کاربران اطلاعات User و SecurityPhase. 1. ثبت کاربران و مراحل امنیتی

خروجی	ورودی	Process
مدل آماده تحلیل	اطلاعات مدل و الگوریتم	2. مدیریت مدل تصمیم‌یار AI
RiskEvent (ثبت شده)	User, SecurityPhase, AI_Model	3. تحلیل و ثبت ریسک‌ها
RiskEvent ReportLog / Dashboard		4. تولید و ارسال گزارش‌ها

جریان داده‌ها:

User → (1) → SecurityPhase → (2) → AI_Model → (3) → RiskEvent → (4) → Dashboard → User

۴.۳ سطح دو (DFD2)

مثال – Process 4: تولید گزارش خودکار

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌های ریسک	RiskEvent	Raw Data	RiskEvent
4.2 تحلیل خودکار داده‌ها با مدل AI	Raw Data	Evaluated Risks	AI_DecisionModel
4.3 تولید گزارش نهایی	Evaluated Risks	Report Summary	ReportLog
4.4 ارسال به داشبورد	Report Summary	Dashboard View	Dashboard, User

جمع‌بندی: DFD

- DFD0: نمای کلی سامانه
- DFD1: فرآیندهای تحلیلی اصلی
- DFD2: تجزیه جزئیات در تولید گزارش و تحلیل ریسک

۵. نمودار وضعیت‌ها (STD – State Transition Diagram)

وضعیت‌ها (States)

Idle

Phase_Assigned

Risk_Analyzed

Report_Generated

Report_Reviewed

از وضعیت	به وضعیت	رویداد / شرط
Idle	Phase_Assigned	تخصیص کاربر به فاز امنیتی
Phase_Assigned	Risk_Analyzed	تحلیل ریسک توسط مدل AI
Risk_Analyzed	Report_Generated	تولید گزارش خودکار
Report_Generated	Report_Reviewed	تأیید گزارش توسط مدیر
Report_Reviewed	Idle	بازگشت به حالت اولیه برای فاز بعدی

۶. نمودار Use Case

Actor	Use Case	ارتباط
Developer	ثبت داده‌های امنیتی	Actor → System
Security Analyst	تحلیل ریسک	Actor → System
System Admin	مدیریت مدل تصمیم‌یار	Actor → System
Project Manager	مشاهده گزارش‌ها و تأیید	Actor ↔ System

موجودیت‌های مرتبط: User, AI_DecisionModel, RiskEvent, ReportLog, Dashboard

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

هدف	نوع ارتباط	ارتباط با	Entity
ورود اطلاعات و مشاهده خروجی ارسال/دریافت			User Smart Security Analyzer
نمایش تحلیل و گزارش‌ها	ارسال	Dashboard	Smart Security Analyzer
اطلاع‌رسانی و تأیید نهایی	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity
ارسال داده‌های پایه	Process 1	User
تعریف مراحل امنیتی		Process 1 SecurityPhase

Entity	ارتباط با	جریان داده
Process 2	AI_Model	آماده سازی مدل تحلیل
Process 3	RiskEvent	ثبت خروجی تحلیل
Process 4	ReportLog Dashboard	تولید گزارش و ارسال به

DFD2

Entity	ارتباط با	جریان داده
RiskEvent	4.1	ورودی داده های خام
	4.1	4.2 انتقال داده های تحلیلی
	4.2	4.3 تولید خروجی ارزیابی
	4.3	4.4 ارسال به داشبورد
	4.4	Dashboard نمایش برای کاربر نهایی

✓ جمع بندی نهایی پروژه ۱۵

بخش	توضیح
DFD0	نمای کلی تعامل کاربران با سیستم تحلیل ریسک امنیتی
DFD1	چهار فرآیند اصلی از ثبت داده تا تولید گزارش
DFD2	جزئیات تولید گزارش خودکار و تحلیل AI
STD	پنج وضعیت از تخصیص فاز تا تأیید گزارش
Use Case	تعامل نقش های مختلف با سیستم (تحلیل گر، مدیر، توسعه دهنده)
User_Phase	کنترل نقش و تخصیص کاربران با تمرکز بر ریسک امنیتی

پروژه ۱۶ - انتخاب سبک معماری بهینه با الگوریتم تصمیمیار چندمعیاره (AHP + ML)

۱. جدول پایه موجودیت ها (Entities)

Entity	Attributes	Type Key	توضیح
User	user_id, username, role, department	INT, VARCHAR, VARCHAR, VARCHAR PK	اطلاعات کاربران (معمار، تحلیل گر، مدیر)

توضیح	Type Key	Attributes	Entity
نیازهای متغیر پروژه یا سیستم	PK INT, VARCHAR, VARCHAR, VARCHAR, VARCHAR	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل نیازها	PK INT, VARCHAR, VARCHAR, VARCHAR	model_id, name, version, description	FunnelModel
ثبت رویدادها و ریسک‌های تحلیل شده	PK INT, INT, INT, VARCHAR, DATE, VARCHAR	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسطه (M:N (User_Requirement)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

نوع خط	ضخامت خط	سطح هوشیاری	رنگ ریسک	نوع ریسک	نوع رابطه	نام رابطه	Entity هدف	Entity منبع	#
نقطه‌چین 2	متوسط	زرد	بالقوه	M:N	تخصیص به نیاز		User Requirement		1
پر 3	بالا	قرمز	بالفعل	1:N	تحلیل توسط مدل		Requirement FunnelModel		2

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه چین

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران
جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

نقش / توضیح موجودیت

وارد کردن داده و دریافت گزارش User

سیستم مرکزی تحلیل ریسک نیازها Smart Funnel System

نمایش شاخص‌ها و گزارش Dashboard

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

Process	ورودی	خروجی
1. ثبت کاربران و نیازها	User, Requirement	داده‌های اولیه User/Requirement
2. مدیریت مدل تصمیم‌یار	FunnelModel	مدل آماده تحلیل
3. تحلیل ریسک User, Requirement, FunnelModel		RiskLog / داده‌های ریسک
4. تولید گزارش و شاخص‌ها	RiskLog	Dashboard / گزارش

جریان:
User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog
→ Process 4 → Dashboard → User

۴.۳ DFD2

جزئیات – Process 4 تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها	Raw Data	KPI Report	User, RiskLog
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

۵. STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Requirement_Added	نیازها ثبت شدند
Analysis_Completed	تحلیل توسط FunnelModel انجام شد
Report_Displayed	گزارش نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها
Requirement_Added	Analysis_Completed	تحلیل توسط FunnelModel
Analysis_Completed	Report_Displayed	نمایش داشبورد
Report_Displayed	Idle	پایان چرخه

۶. Use Case

Actor	Use Case	ارتباط
User	ثبت اطلاعات	Actor → System
User	مشاهده داشبورد	Actor ← System
System Admin	مدیریت مدل Funnel	Actor → System

Actor	Use Case	ارتباط
System Admin	Dashboard مشاهده	Actor → System

موجودیت‌ها / ارتباط‌ها User, System Admin, Dashboard, Use Cases: مرتبط

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

Entity	ارتباط با	نوع ارتباط	جریان داده
User Smart Funnel System		دریافت ارسال/دریافت	ورود داده‌ها و دریافت گزارش
Smart Funnel System	Dashboard	ارسال	نمایش شاخص و گزارش
Dashboard	User	نمایش	ارائه گزارش و شاخص‌ها

DFD1

Entity / Process	ارتباط با	جریان داده
User	Process 1	ارسال اطلاعات
Process 1	User/Requirement	ذخیره داده‌ها
Process 2	FunnelModel	مدیریت مدل تحلیل
Process 3	RiskLog	ثبت ریسک
Process 4	Dashboard	نمایش گزارش

DFD2

Entity / Sub-Process	ارتباط با	جریان داده
RiskLog	4.1	جمع‌آوری داده
4.1	4.2	Raw Data
4.2	4.3	KPI Report
4.3	Dashboard	نمایش

نقش	ارتباط با	Entity
ایجاد و ثبت داده	User Transitions	
مدیریت وضعیت‌ها و فرآیندها	System Admin Transitions	
ثبت وضعیت و اطلاعات	RiskLog States	

Use Case Diagram

جریان داده	Use Case	Actor
ارسال داده‌ها	ثبت داده	User
دریافت گزارش‌ها	مشاهده User Dashboard	
اصلاح و مدیریت داده‌ها و مدل مدیریت مدل Funnel	System Admin	
کنترل شاخص‌ها و گزارش‌ها	مشاهده System Admin Dashboard	

جمع‌بندی نهایی پروژه ۱۶

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب AHP و ML برای پیشنهاد بهترین سبک معماری بر اساس معیارهای کیفی، هزینه و ریسک

پروژه ۱۷ – مدل قیف ریسک برای طراحی سیستم‌های Microservice

۱. جدول پایه موجودیت‌ها و صفات (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (دانشجو، معلم، مدیر)	INT, VARCHAR, VARCHAR, VARCHAR PK	user_id, username, role, department	User

توضیح	Type Key	Attributes	Entity
نیازهای متغیر پروژه یا سیستم	INT, VARCHAR, VARCHAR, VARCHAR, PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل نیازها	INT, VARCHAR, VARCHAR, VARCHAR, PK	model_id, name, version, description	FunnelModel
ثبت رویدادها و ریسک‌های تحلیل شده	INT, INT, INT, VARCHAR, DATE, VARCHAR, PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسط (M:N (User_Requirement)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – کاربر ممکن است دسترسی به نیاز صحیح نداشته باشد	PK	INT	شناسه یکتا هر رکورد (User-Requirement)	ur_id
ارجاع نادرست – داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
ارجاع اشتباه – نیازهای واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
نقش اشتباه – تصمیمات تحلیل ممکن است اشتباه ثبت شود		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

✓ ویژگی‌های خاص جدول واسط:

- مدیریت ارتباط M:N بین User و Requirement
- ریسک‌ها روی صحت داده‌ها و نقش‌ها تمرکز دارند
- در Draw.io سه بخش دارد: نام جدول / فیلدها / ریسک‌ها
- جزئیات DFD با رنگ و ضخامت خطوط برای ریسک نمایش داده می‌شوند

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User Requirement		تخصیص به نیاز	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	Requirement FunnelModel		تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا	3	پر
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

نکات برای: Draw.io

- هر موجودیت و جدول واسط سه بخش دارد: نام / فیلدها / ریسک‌ها
- روابط بین موجودیت‌ها ضخامت و نوع خط بر اساس نوع ریسک نمایش داده می‌شود
- رنگ ریسک‌ها: سبز = کم، زرد = متوسط، قرمز = بالا
- خطوط: پر = بالفعل، نقطه‌چین = بالقوه

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ سطح صفر (DFD0)

هدف: نمایش تعامل سیستم با کاربران و محیط بیرونی به صورت کلی
جریان داده‌ها:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

موجودیت‌ها در: DFD0:

نقش / توضیح	موجودیت
دانشجو، معلم یا مدیر – وارد کردن داده‌ها و دریافت گزارش	User
سیستم مرکزی تحلیل ریسک نیازها با مدل قیف تصمیم‌یار	Smart Funnel System
نمایش شاخص‌ها و گزارش‌های تحلیل نیاز	Dashboard

۴.۲ سطح یک (DFD1)

هدف: نمایش فرآیندهای اصلی و جریان داده‌ها

فرآیندها:

۱. ثبت کاربران و نیازها
۲. مدیریت مدل تصمیم‌یار هوشمند
۳. تحلیل ریسک و ثبت رویدادها
۴. تولید گزارش و شاخص‌ها

جریان داده‌ها:

User → Process 1 → Requirement / User → Process 2 → FunnelModel → Process 3 → RiskLog → Process 4 → Dashboard → User

جدول فرآیندها و جریان داده:

Process	ورودی	خروجی
1. ثبت کاربران و نیازها	User/Requirement اطلاعات	داده‌های اولیه
2. مدیریت مدل تصمیم‌یار	FunnelModel اطلاعات	مدل آماده تحلیل
3. تحلیل ریسک	User, Requirement, FunnelModel	RiskLog / داده‌های ریسک
4. تولید گزارش	RiskLog	Dashboard / گزارش و شاخص‌ها

DFD1: User, Requirement, FunnelModel, RiskLog, Dashboard در موجودیت‌ها

۴.۳ سطح دو (DFD2)

هدف: شکستن فرآیندها برای تحلیل دقیق

مثال – Process 4: تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها	Raw Data	KPI Report	User, RiskLog
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

DFD2: RiskLog, 4.1, 4.2, 4.3, Dashboard, User در موجودیت‌ها

جمع‌بندی: DFD 

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها

• DFD2: جزئیات دقیق فرآیندها

۵. نمودار STD – State Transition Diagram

States: Idle, User_Registered, Requirement_Added, Analysis_Completed, Report_Displayed

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها
Requirement_Added	Analysis_Completed	تحلیل توسط FunnelModel
Analysis_Completed	Report_Displayed	نمایش داشبورد
Report_Displayed	Idle	پایان چرخه

STD: User, System Admin, RiskLog, FunnelModel, States, Transitions موجودیت‌های مرتبط

۶. نمودار Use Case

Actor	Use Case	ارتباط
User	ثبت اطلاعات Actor → System	
User	مشاهده Dashboard Actor → System	
System Admin	مدیریت مدل Funnel Actor → System	
System Admin	مشاهده Dashboard Actor → System	

موجودیت‌ها / ارتباط‌ها: User, System Admin, Dashboard Use Cases مرتبط

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0:

Entity	نوع ارتباط	ارتباط با	جریان داده / هدف
User Smart Funnel System	دریافت / ارسال		ارسال اطلاعات و دریافت گزارش
Smart Funnel System	ارسال	Dashboard	نمایش شاخص و گزارش

Entity	ارتباط با	نوع ارتباط	جریان داده / هدف
Dashboard	User	نمایش	ارائه گزارش و شاخص‌ها

DFD1:

Entity	ارتباط با	جریان داده
User	Process 1	ارسال اطلاعات
Process 1	User/Requirement	ذخیره داده‌ها
Process 2	FunnelModel	مدیریت مدل تحلیل
Process 3	RiskLog	ثبت ریسک
Process 4	Dashboard	نمایش گزارش

DFD2:

Entity	ارتباط با	جریان داده
RiskLog	4.1	جمع‌آوری داده
4.1	4.2	Raw Data
4.2	4.3	KPI Report
4.3	Dashboard	نمایش

STD:

Entity	ارتباط با	نقش
User Transitions		عامل ایجاد و ثبت داده
System Admin Transitions		عامل تغییر وضعیت و مدیریت
RiskLog	States	ثبت وضعیت و اطلاعات

Use Case Diagram:

Actor	Use Case	جریان داده
User	ثبت داده	ارسال داده‌ها به سیستم
User	مشاهده Dashboard	دریافت گزارش‌ها
System Admin	Funnel	اصلاح و مدیریت داده‌ها و مدل مدیریت مدل

Actor	Use Case	جریان داده
System Admin	مشاهده Dashboard	کنترل شاخص‌ها و گزارش‌ها

جمع‌بندی نهایی پروژه ۱۷

- **DFD0:** نمای کلی و جعبه سیاه
- **DFD1:** فرآیندهای اصلی + جریان داده‌ها
- **DFD2:** جزئیات دقیق فرآیندها
- **STD:** وضعیت‌ها و رویدادها
- **Use Case:** تعامل Actor ها و سیستم
- **User Requirement:** جدول واسط دارای فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io

پروژه ۱۸ – پیش‌بینی کیفیت معماری بر اساس Code Metrics و ریسک طراحی

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	INT, VARCHAR, VARCHAR, PK, VARCHAR	user_id, username, role, department	User
شاخص‌های کد (مانند LOC, Cyclomatic Complexity) برای هر پروژه	INT, VARCHAR, VARCHAR, FLOAT, PK, INT	metric_id, name, description, value, project_id	CodeMetrics
اطلاعات سبک و ساختار معماری پروژه	INT, VARCHAR, VARCHAR, INT, PK, INT	arch_id, name, style, complexity, project_id	Architecture
ثبت رویدادها و ریسک‌های طراحی مرتبط با کد و معماری	INT, INT, INT, VARCHAR, DATE, PK, VARCHAR	log_id, arch_id, metric_id, severity, date, notes	RiskLog
مدل پیش‌بینی کیفیت معماری با استفاده از Code Metrics و ریسک	INT, VARCHAR, FLOAT, VARCHAR, PK, VARCHAR	model_id, type, accuracy, parameters, version	QualityModel

۲. جدول واسطه M:N (User_Architecture)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گم‌شده – دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ua_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
معماری واقعی با تحلیل منطبق نیست	FK → Architecture(arch_id)	INT	ارجاع به جدول Architecture	arch_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند	-	VARCHAR	نقش کاربر در ارزیابی معماری (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Architecture ، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Architecture	تخصیص به معماری	M:N	بالتوجه	زرد	متوسط	2	نقطه‌چین
2	Architecture	CodeMetrics	جمع‌آوری شاخص‌ها	1:N	بالفعل	قرمز	بالا	3	پر
3	CodeMetrics	QualityModel	آموزش مدل پیش‌بینی	1:N	بالتوجه	زرد	متوسط	2	نقطه‌چین
4	QualityModel	RiskLog	تحلیل ریسک طراحی	1:N	بالفعل	قرمز	بالا	3	پر
5	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر

۴. نمودارهای فرآیند و رفتار سیستم

هدف: تعامل کلی سیستم با کاربران

جریان داده:

[User] → [Quality Prediction System] → [Dashboard / Reports] → [User]

نقش / توضیح موجودیت

User وارد کردن داده و دریافت گزارش

سیستم پیش‌بینی کیفیت و تحلیل ریسک طراحی Quality Prediction System

Dashboard نمایش شاخص‌ها و گزارش کیفیت معماری

DFD1 ۴.۲

فرآیندها و جریان داده‌ها:

Process	ورودی	خروجی
1. ثبت کاربران و معماری	User, Architecture	User/Architecture اولیه‌های
2. جمع‌آوری Code Metrics	Architecture	CodeMetrics
3. آموزش مدل پیش‌بینی کیفیت	CodeMetrics, Architecture	QualityModel
4. تحلیل ریسک و تولید گزارش	QualityModel, CodeMetrics	RiskLog / Dashboard

جریان داده:

User → Process 1 → Architecture/User → Process 2 → CodeMetrics → Process 3 → QualityModel → Process 4 → RiskLog/Dashboard → User

DFD2 ۴.۳

جزئیات Process 4 – تحلیل ریسک و تولید گزارش

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog, CodeMetrics	CombinedData	RiskLog, CodeMetrics
4.2 پیش‌بینی کیفیت	CombinedData	QualityScore	Architecture, QualityModel
4.3 تولید گزارش و داشبورد	QualityScore, RiskLog	Dashboard Report	Dashboard, User

۵ . STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Architecture_Added	معماری‌ها ثبت شدند
Metrics_Collected	Code Metrics جمع‌آوری شد
Model_Trained	مدل پیش‌بینی کیفیت آموزش داده شد
Analysis_Completed	تحلیل کیفیت و ریسک انجام شد
Report_Displayed	گزارش و داشبورد نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Architecture_Added	ثبت معماری‌ها
Architecture_Added	Metrics_Collected	جمع‌آوری شاخص‌ها
Metrics_Collected	Model_Trained	آموزش مدل QualityModel
Model_Trained	Analysis_Completed	پیش‌بینی کیفیت و تحلیل ریسک
Analysis_Completed	Report_Displayed	نمایش گزارش
Report_Displayed	Idle	پایان چرخه

۶ . Use Case

Actor	Use Case	ارتباط
User	Actor → System	ثبت معماری و داده‌ها
User	Actor ← System	مشاهده Dashboard
System Admin	Actor → System	مدیریت مدل QualityModel

Actor	Use Case	ارتباط
System Admin	رиск و کیفیت گزارش مشاهده	Actor ← System
موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, CodeMetrics, Architecture, RiskLog, QualityModel		

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت گزارش ارسال/دریافت User Quality Prediction System			
نمایش شاخص‌ها و گزارش	ارسال	Dashboard	Quality Prediction System
ارائه گزارش و شاخص‌ها	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها	Process 1	User
ذخیره داده‌ها	Architecture/User	Process 1
جمع‌آوری شاخص‌ها	CodeMetrics	Process 2
آموزش مدل پیش‌بینی	QualityModel	Process 3
تولید گزارش	RiskLog/Dashboard	Process 4

DFD2

جریان داده	ارتباط با	Entity / Sub-Process
جمع‌آوری داده‌ها	4.1	RiskLog, CodeMetrics
4.2 CombinedData	4.1	
QualityScore	4.2	4.3
نمایش	4.3	Dashboard

STD

نقش	ارتباط با	Entity
ایجاد و ثبت داده	User Transitions	
مدیریت وضعیت‌ها و مدل	System Admin Transitions	
ثبت وضعیت و اطلاعات	States	RiskLog, QualityModel

Use Case Diagram

جریان داده	Use Case	Actor
ارسال داده‌ها	ثبت داده	User
دریافت گزارش‌ها	مشاهده Dashboard	User
اصلاح و مدیریت داده‌ها	مدیریت مدل QualityModel	System Admin
کنترل شاخص‌ها و گزارش‌ها مشاهده Dashboard		System Admin

جمع‌بندی نهایی پروژه ۱۸

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- User_Architecture: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند برای پیش‌بینی کیفیت معماری پروژه با استفاده از Code Metrics و تحلیل ریسک طراحی، به منظور بهبود کیفیت، کاهش ریسک و بهینه‌سازی معماری

پروژه ۱۹ – تحلیل تعامل بین مؤلفه‌ها و شناسایی نقاط بحرانی با مدل قیف وابستگی

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department INT, VARCHAR, VARCHAR, VARCHAR	User
مؤلفه‌های سیستم و میزان بحرانی بودن آنها	PK	comp_id, name, type, criticality INT, VARCHAR, VARCHAR, VARCHAR	Component
مدل قیف وابستگی برای تحلیل تعامل مؤلفه‌ها	PK	model_id, name, version, description INT, VARCHAR, VARCHAR, VARCHAR	FunnelModel
ثبت تعامل‌ها و ریسک‌های بحرانی بین مؤلفه‌ها	PK	log_id, comp_id, model_id, severity, date, notes INT, INT, INT, VARCHAR, DATE, VARCHAR	DependencyLog

۲. جدول واسطه (M:N (User_Component)

ریسک (نام و توضیح) کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گم‌شده – دسترسی نادرست کاربر	INT	شناسه یکتا هر رکورد	uc_id
داده‌ها با کاربر واقعی مطابقت ندارد	INT	ارجاع به جدول User	user_id
مؤلفه واقعی با تحلیل منطبق نیست	INT	ارجاع به جدول Component	comp_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند	VARCHAR	نقش کاربر در تحلیل وابستگی (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Component، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Component	تخصیص به مؤلفه	M:N	بالقوه	زرد	متوسط		نقطه چین 2
2	Component	FunnelModel	تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا		پر 3
3	User	DependencyLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا		پر 3
4	Component	DependencyLog	مرتبط با مؤلفه	1:N	بالقوه	زرد	متوسط		نقطه چین 2

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران
جریان داده:

[User] → [Dependency Funnel System] → [Dashboard / Reports] → [User]

موجودیت

نقش / توضیح

User

وارد کردن داده و دریافت گزارش

سیستم تحلیل وابستگی مؤلفه‌ها و شناسایی نقاط بحرانی Dependency Funnel System

Dashboard

نمایش شاخص‌ها و گزارش نقاط بحرانی

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

ورودی	Process	خروجی
User, Component	1. ثبت کاربران و مؤلفه‌ها	داده‌های اولیه User/Component
FunnelModel	2. مدیریت مدل قیف وابستگی	مدل آماده تحلیل
User, Component, FunnelModel	3. تحلیل تعامل و ریسک	DependencyLog / داده‌های ریسک
DependencyLog	4. تولید گزارش و شاخص‌ها	Dashboard / گزارش

جریان داده:

User → Process 1 → Component/User → Process 2 → FunnelModel → Process 3 →
DependencyLog → Process 4 → Dashboard → User

DFD2 ۴.۳

جزئیات – Process 4 تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	DependencyLog	Raw Data	DependencyLog
4.2 محاسبه شاخص‌های بحرانی	Raw Data	KPI Report	Component, DependencyLog
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

۵ STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Component_Added	مؤلفه‌ها ثبت شدند
Analysis_Completed	تحلیل توسط FunnelModel انجام شد
Report_Displayed	گزارش نقاط بحرانی نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Component_Added	ثبت مؤلفه‌ها
Component_Added	Analysis_Completed	تحلیل تعامل و شناسایی نقاط بحرانی
Analysis_Completed	Report_Displayed	نمایش داشبورد
Report_Displayed	Idle	پایان چرخه

ارتباط	Use Case	Actor
Actor → System	ثبت مؤلفه و داده‌ها	User
Actor ← System	مشاهده Dashboard نقاط بحرانی	User
Actor → System	مدیریت مدل Funnel	System Admin
Actor ← System	مشاهده گزارش بحرانی	System Admin

موجودیت‌ها / ارتباط‌ها, DependencyLog, Component, Dashboard, System Admin, User :
FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها 

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت گزارش	ارسال/دریافت	Dependency Funnel System	User
نمایش شاخص‌ها و نقاط بحرانی	ارسال	Dashboard	Dependency Funnel System
ارائه گزارش و شاخص‌ها	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها	Process 1	User
ذخیره داده‌ها		Process 1 Component/User
مدیریت مدل تحلیل	FunnelModel	Process 2
ثبت ریسک و وابستگی‌ها	DependencyLog	Process 3
نمایش گزارش	Dashboard	Process 4

DFD2

Entity / Sub-Process	ارتباط با	جریان داده
DependencyLog	4.1	جمع‌آوری داده‌ها
4.1	4.2	Raw Data
4.2	4.3	KPI Report
4.3 Dashboard		نمایش

STD

Entity	ارتباط با	نقش
User Transitions		ایجاد و ثبت داده
System Admin Transitions		مدیریت وضعیت‌ها و مدل
DependencyLog	States	ثبت وضعیت و اطلاعات

Use Case Diagram

Actor	Use Case	جریان داده
User	ثبت داده	ارسال داده‌ها
User Dashboard	مشاهده	دریافت گزارش‌ها
System Admin Funnel	مدیریت مدل	اصلاح و مدیریت داده‌ها
System Admin Dashboard	مشاهده	کنترل شاخص‌ها و گزارش‌ها

جمع‌بندی نهایی پروژه ۱۹ ✓

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- User_Component: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با مدل قیف وابستگی برای تحلیل تعامل بین مؤلفه‌ها و شناسایی نقاط بحرانی، به منظور بهبود کیفیت معماری، کاهش ریسک و بهینه‌سازی طراحی

پروژه ۲۰ - سیستم هوشمند پیشنهاد سبک معماری برای پروژه‌های نرم‌افزاری جدید با الگوریتم ML

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department	User
نیازهای متغیر پروژه یا سیستم	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل نیازها و پیشنهاد سبک معماری	PK	model_id, name, version, description	FunnelModel
ثبت رویدادها و ریسک‌های تحلیل شده	PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسطه (M:N (User_Requirement

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده - دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User Requirement		تخصیص به نیاز	M:N	بالقوه	زرد	متوسط	2	نقطه چین
2	Requirement FunnelModel		تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا	3	پر
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه چین

۴. نمودارهای فرآیند و رفتار سیستم

DFD0 ۴.۱

هدف: تعامل کلی سیستم با کاربران
جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

نقش / توضیح موجودیت

User وارد کردن داده و دریافت پیشنهاد سبک معماری

Smart Funnel System سیستم تحلیل ریسک نیازها و پیشنهاد سبک معماری

Dashboard نمایش شاخص‌ها و سبک پیشنهادی

DFD1 ۴.۲

فرآیندها و جریان داده‌ها:

Process	ورودی	خروجی
1. ثبت کاربران و نیازها	User, Requirement	داده‌های اولیه User/Requirement
2. مدیریت مدل تصمیم‌یار	FunnelModel	مدل آماده تحلیل و پیشنهاد سبک

خروجی	ورودی	Process
RiskLog / پیشنهاد سبک معماری	User, Requirement, FunnelModel	3. تحلیل ریسک و پیشنهاد سبک
Dashboard / گزارش و پیشنهادات RiskLog		4. تولید گزارش و شاخص‌ها

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog → Process 4 → Dashboard → User

۴.۳ DFD2

جزئیات – Process 4 تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها و پیش‌بینی سبک	Raw Data	KPI & ML Report	User, RiskLog
4.3 ارسال به داشبورد	KPI & ML Report	Dashboard Display	Dashboard, User

۵ STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Requirement_Added	نیازها ثبت شدند
Analysis_Completed	تحلیل توسط FunnelModel انجام شد و سبک پیشنهادی ایجاد شد
Report_Displayed	گزارش و پیشنهاد سبک معماری نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها

از وضعیت	به وضعیت	رویداد / شرط
Requirement_Added	Analysis_Completed	تحلیل توسط + FunnelModel الگوریتم ML
Analysis_Completed	Report_Displayed	نمایش داشبورد
Report_Displayed	Idle	پایان چرخه

۶. Use Case

Actor	Use Case	ارتباط
User	Actor → System	ثبت نیازها و داده‌ها
User	Actor ← System	مشاهده Dashboard و پیشنهاد سبک
System Admin	Actor → System	مدیریت مدل Funnel + ML
System Admin	Actor ← System	مشاهده گزارش و شاخص‌ها

موجودیت‌ها / ارتباطها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت پیشنهاد سبک ارسال/دریافت			User Smart Funnel System
نمایش شاخص‌ها و پیشنهاد سبک	ارسال		Smart Funnel System
ارائه گزارش و شاخص‌ها	نمایش		Dashboard
		User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها	Process 1	User
ذخیره داده‌ها		Process 1 Requirement/User
مدیریت مدل تحلیل و ML	FunnelModel	Process 2

Entity / Process	ارتباط با	جریان داده
Process 3	RiskLog	ثبت ریسک‌ها و پیشنهاد سبک
Process 4	Dashboard	نمایش گزارش و پیشنهاد سبک

DFD2

Entity / Sub-Process	ارتباط با	جریان داده
RiskLog	4.1	جمع‌آوری داده
4.1	4.2	Raw Data
4.2	4.3	KPI & ML Report
4.3 Dashboard		نمایش

STD

Entity	ارتباط با	نقش
User Transitions		ایجاد و ثبت داده
System Admin Transitions		مدیریت وضعیت‌ها و مدل
RiskLog	States	ثبت وضعیت و اطلاعات

Use Case Diagram

Actor	Use Case	جریان داده
User	ثبت داده	ارسال داده‌ها
User	مشاهده Dashboard و سبک پیشنهادی	دریافت گزارش و پیشنهاد
System Admin	Funnel + ML	اصلاح و مدیریت داده‌ها و مدل مدیریت مدل
System Admin	Dashboard	کنترل شاخص‌ها و گزارش‌ها مشاهده

جمع‌بندی نهایی پروژه ۲۰ 

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها

- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب الگوریتم ML و مدل Funnel برای پیشنهاد بهترین سبک معماری بر اساس معیارهای کیفی، هزینه، ریسک و پیش‌بینی داده‌های پروژه‌های نرم‌افزاری جدید

پروژه ۲۱ – بهینه‌سازی ساختار لایه‌ای سیستم با الگوریتم یادگیری خودکار (Auto Layering)

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department	User
نیازهای متغیر پروژه یا سیستم	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل نیازها و پیشنهاد ساختار لایه‌ای	PK	model_id, name, version, description	FunnelModel
ثبت رویدادها و ریسک‌های تحلیل شده	PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسط (M:N) (User_Requirement)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id

رиск (نام و توضیح)	کلید	نوع داده	توضیح	ستون
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط - کند		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role
ویژگی‌ها: مدیریت M:N بین User و Requirement ، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)				

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Requirement	تخصیص به نیاز	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	Requirement FunnelModel		تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا	3	پر
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران
جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

نقش / توضیح موجودیت

User وارد کردن داده و دریافت گزارش لایه‌بندی پیشنهادی

Smart Funnel System Auto Layering سیستم تحلیل ریسک نیازها و بهینه‌سازی لایه‌بندی با الگوریتم

Dashboard نمایش شاخص‌ها و ساختار لایه‌ای پیشنهادی

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

خروجی	ورودی	Process
داده‌های اولیه User/Requirement	User, Requirement	1. ثبت کاربران و نیازها
مدل آماده تحلیل و Auto Layering	FunnelModel	2. مدیریت مدل تصمیم‌یار
RiskLog / ساختار لایه‌ای پیشنهادی	User, Requirement, FunnelModel	3. تحلیل ریسک و بهینه‌سازی لایه‌ها
Dashboard / گزارش و پیشنهادات RiskLog		4. تولید گزارش و شاخص‌ها

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog → Process 4 → Dashboard → User

۴.۳ DFD2

جزئیات – Process 4 تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها و بهینه‌سازی لایه‌ها	Raw Data	KPI & Layering Report	User, RiskLog
4.3 ارسال به داشبورد	KPI & Layering Report	Dashboard Display	Dashboard, User

۵ STD – State Transition Diagram

توضیح	States
سیستم منتظر داده است	Idle
کاربران ثبت شدند	User_Registered
نیازها ثبت شدند	Requirement_Added
تحلیل توسط FunnelModel و الگوریتم Auto Layering انجام شد	Analysis_Completed
گزارش و ساختار لایه‌ای پیشنهادی نمایش داده شد	Report_Displayed

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها
Requirement_Added	Analysis_Completed	تحلیل توسط FunnelModel + Auto Layering
Analysis_Completed	Report_Displayed	نمایش داشبورد
Report_Displayed	Idle	پایان چرخه

۶. Use Case

Actor	Use Case	ارتباط
User	Actor → System	ثبت نیازها و داده‌ها
User	Actor ← System	مشاهده Dashboard و لایه‌بندی پیشنهادی
System Admin	Actor → System	مدیریت مدل Funnel + Auto Layering
System Admin	Actor ← System	مشاهده گزارش و شاخص‌ها

موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت لایه‌بندی پیشنهادی	ارسال/دریافت	Smart Funnel System	User
نمایش شاخص‌ها و ساختار لایه‌ای پیشنهادی	ارسال	Dashboard	Smart Funnel System
ارائه گزارش و شاخص‌ها	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها	Process 1	User
ذخیره داده‌ها	Process 1 Requirement/User	
مدیریت مدل تحلیل و Auto Layering	FunnelModel	Process 2
ثبت ریسک‌ها و ساختار لایه‌ای پیشنهادی	RiskLog	Process 3
نمایش گزارش و ساختار لایه‌ای	Dashboard	Process 4

DFD2

جریان داده	ارتباط با	Entity / Sub-Process
جمع‌آوری داده	4.1	RiskLog
Raw Data	4.2	4.1
4.3 KPI & Layering Report	4.3	4.2
نمایش	4.3 Dashboard	4.3

STD

نقش	ارتباط با	Entity
ایجاد و ثبت داده	User Transitions	
مدیریت وضعیت‌ها و مدل	System Admin Transitions	
ثبت وضعیت و اطلاعات	States	RiskLog

Use Case Diagram

جریان داده	Use Case	Actor
ارسال داده‌ها	ثبت داده	User
دریافت گزارش و پیشنهاد	مشاهده Dashboard و لایه‌بندی پیشنهادی	User
اصلاح و مدیریت داده‌ها و مدل مدیریت مدل Funnel + Auto Layering		System Admin
کنترل شاخص‌ها و گزارش‌ها	مشاهده Dashboard	System Admin

- DFD0: نمودار کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب الگوریتم Auto Layering و Funnel برای پیشنهاد و بهینه‌سازی ساختار لایه‌ای سیستم‌ها بر اساس معیارهای کیفی، هزینه و ریسک

پروژه ۲۲ Refactoring – هوشمند معماری برای کاهش Coupling و افزایش Cohesion

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department	User
نیازهای متغیر پروژه یا سیستم	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل نیازها و پیشنهاد Refactoring	PK	model_id, name, version, description	FunnelModel
ثبت رویدادها و ریسک‌های تحلیل شده	PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسط (M:N (User_Requirement)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق → FK Requirement(req_id)		INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Requirement	تخصیص به نیاز	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	Requirement FunnelModel		تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا	3	پر
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران و دریافت Refactoring پیشنهادی
جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

نقش / توضیح موجودیت

User وارد کردن داده‌ها و دریافت گزارش Refactoring

سیستم تحلیل ریسک نیازها و پیشنهاد Refactoring برای کاهش Coupling و افزایش Cohesion
Smart Funnel System

DFD1 ۴.۲

فرآیندها و جریان داده‌ها:

خروجی	ورودی	Process
داده‌های اولیه User/Requirement	User, Requirement	1. ثبت کاربران و نیازها
مدل آماده تحلیل Refactoring	FunnelModel	2. مدیریت مدل تصمیم‌یار
RiskLog / ساختار معماری بهینه	User, Requirement, FunnelModel	3. تحلیل ریسک و پیشنهاد Refactoring
Dashboard / گزارش و پیشنهادات	RiskLog	4. تولید گزارش و شاخص‌ها

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog
→ Process 4 → Dashboard → User

DFD2 ۴.۳

جزئیات – Process 4 تولید گزارش و شاخص‌ها

Related Entities	Output	Input	Sub-Process
RiskLog	Raw Data	RiskLog	4.1 جمع‌آوری داده‌ها
User, RiskLog	KPI & Refactoring Report	Raw Data	4.2 محاسبه شاخص‌ها و پیشنهاد Refactoring
Dashboard, User	Dashboard Display	KPI & Refactoring Report	4.3 ارسال به داشبورد

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Requirement_Added	نیازها ثبت شدند
Analysis_Completed	تحلیل توسط FunnelModel و الگوریتم Refactoring انجام شد
Report_Displayed	گزارش و پیشنهاد Refactoring نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها
Requirement_Added	Analysis_Completed	تحلیل توسط FunnelModel + الگوریتم Refactoring
Analysis_Completed	Report_Displayed	نمایش داشبورد
Report_Displayed	Idle	پایان چرخه

۶. Use Case

Actor	ارتباط	Use Case
User	Actor → System	ثبت نیازها و داده‌ها
User	Actor ← System	مشاهده Dashboard و پیشنهاد Refactoring
System Admin	Actor → System	مدیریت مدل + Funnel الگوریتم Refactoring
System Admin	Actor ← System	مشاهده گزارش و شاخص‌ها

موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

Entity	ارتباط با	نوع ارتباط	جریان داده
User	Smart Funnel System	ارسال/دریافت	ورود داده‌ها و دریافت Refactoring پیشنهادی
Smart Funnel System	Dashboard	ارسال	نمایش شاخص‌ها و پیشنهاد Refactoring
Dashboard	User	نمایش	ارائه گزارش و شاخص‌ها

DFD1

Entity / Process	ارتباط با	جریان داده
User	Process 1	ارسال داده‌ها
Process 1 Requirement/User		ذخیره داده‌ها
Process 2	FunnelModel	مدیریت مدل تحلیل و Refactoring
Process 3	RiskLog	ثبت ریسک‌ها و پیشنهاد معماری
Process 4	Dashboard Refactoring	نمایش گزارش و پیشنهاد

DFD2

Entity / Sub-Process	ارتباط با	جریان داده
RiskLog	4.1	جمع‌آوری داده
4.1	4.2	Raw Data
4.2	4.3 KPI & Refactoring Report	
4.3 Dashboard		نمایش

STD

Entity	ارتباط با	نقش
User Transitions		ایجاد و ثبت داده
System Admin Transitions		مدیریت وضعیت‌ها و مدل
RiskLog	States	ثبت وضعیت و اطلاعات

Use Case Diagram

جریان داده	Use Case	Actor
ارسال داده‌ها	ثبت داده	User
دریافت گزارش و پیشنهاد	مشاهده Dashboard و پیشنهاد Refactoring	User
اصلاح و مدیریت داده‌ها و مدل مدیریت مدل + Funnel الگوریتم Refactoring		System Admin
کنترل شاخص‌ها و گزارش‌ها	مشاهده Dashboard	System Admin

✓ جمع‌بندی نهایی پروژه ۲۲

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب الگوریتم Refactoring و Funnel برای کاهش Coupling، افزایش Cohesion و بهینه‌سازی معماری سیستم‌ها بر اساس معیارهای کیفی، هزینه و ریسک

پروژه ۲۳ - مدل قیف ارزیابی کیفیت معماری با معیارهای عملکرد، امنیت، مقیاس‌پذیری

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department	User
نیازهای متغیر پروژه یا سیستم	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل کیفیت معماری	PK	model_id, name, version, description	FunnelModel

توضیح	Type Key	Attributes	Entity
ثبت رویدادها و ریسک‌های تحلیل شده	PK	log_id, req_id, model_id, severity, date, notes	RiskLog
		INT, INT, INT, VARCHAR, DATE, VARCHAR	

۲. جدول واسطه (User_Requirement) M:N

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

نوع خط	ضخامت خط	سطح هوشیاری	رنگ ریسک	نوع ریسک	نوع رابطه	نام رابطه	Entity هدف	Entity منبع	#
نقطه‌چین 2	متوسط	زرد	بالقوه	M:N	تخصیص به نیاز	User Requirement	1		
پر 3	بالا	قرمز	بالفعل	1:N	تحلیل توسط مدل	Requirement FunnelModel	2		
پر 3	بالا	قرمز	بالفعل	1:N	ثبت ریسک	RiskLog	User	3	
نقطه‌چین 2	متوسط	زرد	بالقوه	1:N	مرتبط با نیاز	RiskLog	Requirement	4	

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران و ارزیابی کیفیت معماری

جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

موجودیت

نقش / توضیح

User

وارد کردن داده‌ها و دریافت گزارش کیفیت معماری

سیستم مرکزی تحلیل کیفیت معماری با معیارهای عملکرد، امنیت و مقیاس‌پذیری Smart Funnel System

Dashboard

نمایش شاخص‌ها و گزارش‌های کیفیت و رتبه‌بندی معماری

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

Process	ورودی	خروجی
1. ثبت کاربران و نیازها	User, Requirement	داده‌های اولیه User/Requirement
2. مدیریت مدل تصمیم‌یار	FunnelModel	مدل آماده تحلیل کیفیت
3. تحلیل کیفیت معماری	User, Requirement, FunnelModel	RiskLog / داده‌های کیفیت و شاخص‌ها
4. تولید گزارش و شاخص‌ها	RiskLog	Dashboard / گزارش KPI و شاخص‌ها

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog
→ Process 4 → Dashboard → User

۴.۳ DFD2

جزئیات - Process 4 تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها (Performance, Security, Scalability)	Raw Data	KPI Report	User, RiskLog

Sub-Process	Input	Output	Related Entities
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

۵. STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Requirement_Added	نیازها ثبت شدند
Analysis_Completed	تحلیل کیفیت معماری توسط FunnelModel انجام شد
Report_Displayed	گزارش و KPI نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها
Requirement_Added	Analysis_Completed	تحلیل توسط FunnelModel
Analysis_Completed	Report_Displayed	نمایش داشبورد و KPI
Report_Displayed	Idle	پایان چرخه

۶. Use Case

Actor	Use Case	ارتباط
User	Actor → System	ثبت نیازها و داده‌ها
User	Actor ← System	مشاهده Dashboard و KPI
System Admin	Actor → System	مدیریت مدل Funnel و معیارهای کیفیت

Actor	Use Case	ارتباط
System Admin	Actor ← System مشاهده گزارش و شاخص‌ها	
موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel		

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

Entity	ارتباط با	نوع ارتباط	جریان داده
ورود داده‌ها و دریافت گزارش KPI ارسال/دریافت User Smart Funnel System			
Smart Funnel System	Dashboard	ارسال	نمایش شاخص‌ها و KPI
Dashboard	User	نمایش	ارائه گزارش و شاخص‌ها

DFD1

Entity / Process	ارتباط با	جریان داده
User	Process 1	ارسال داده‌ها
Process 1	User/Requirement	ذخیره داده‌ها
Process 2	FunnelModel	مدیریت مدل تحلیل کیفیت
Process 3	RiskLog	ثبت ریسک‌ها و شاخص‌های کیفیت
Process 4	Dashboard	نمایش گزارش و KPI

DFD2

Entity / Sub-Process	ارتباط با	جریان داده
RiskLog	4.1	جمع‌آوری داده
4.1	4.2	Raw Data
4.2	4.3	KPI Report
4.3 Dashboard		نمایش

STD

نقش	ارتباط با	Entity
ایجاد و ثبت داده	User Transitions	
مدیریت وضعیت‌ها و مدل	System Admin Transitions	
ثبت وضعیت و اطلاعات	RiskLog States	

Use Case Diagram

جریان داده	Use Case	Actor
ارسال داده‌ها	ثبت داده	User
دریافت گزارش‌ها	مشاهده KPI و Dashboard	User
اصلاح و مدیریت داده‌ها و معیارهای کیفیت مدیریت مدل	Funnel	System Admin
کنترل شاخص‌ها و گزارش‌ها	مشاهده Dashboard	System Admin

جمع‌بندی نهایی پروژه ۲۳

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب AHP و ML برای ارزیابی کیفیت معماری بر اساس معیارهای عملکرد، امنیت و مقیاس‌پذیری، همراه با تحلیل ریسک و ارائه گزارش KPI

پروژه ۲۴ – تحلیل و شبیه‌سازی تغییرات معماری در محیط DevOps و CI/CD

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department	User
نیازهای متغیر پروژه یا سیستم	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل تغییرات معماری	PK	model_id, name, version, description	FunnelModel
ثبت رویدادها و ریسک‌های مرتبط با تغییرات	PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسطه (M:N (User_Requirement)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User Requirement		تخصیص به نیاز	M:N	بالقوه	زرد	متوسط	2	نقطه چین
2	Requirement FunnelModel		تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا	3	پر
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه چین

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران در محیط DevOps / CI-CD و مدیریت تغییرات

جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

موجودیت

نقش / توضیح

User

وارد کردن داده‌های تغییرات معماری و دریافت گزارش

Smart Funnel
System

سیستم مرکزی تحلیل تغییرات معماری، ارزیابی ریسک و شبیه‌سازی در محیط DevOps

Dashboard

نمایش شاخص‌ها، وضعیت CI/CD، تغییرات و ریسک‌ها

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

Process	ورودی	خروجی
1. ثبت کاربران و نیازها	User, Requirement	داده‌های اولیه User/Requirement
2. مدیریت مدل تصمیم‌یار	FunnelModel	مدل آماده تحلیل تغییرات معماری
3. تحلیل تغییرات و ریسک	User, Requirement, FunnelModel	RiskLog / داده‌های ریسک و شبیه‌سازی
4. تولید گزارش و شاخص‌ها	RiskLog	Dashboard / گزارش تغییرات و KPI

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog
→ Process 4 → Dashboard → User

DFD2 ۴.۳

جزئیات – Process 4 تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها (تغییرات، کیفیت CI/CD، ریسک)	Raw Data	KPI Report	User, RiskLog
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

۵ . STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Requirement_Added	نیازها و تغییرات ثبت شدند
Analysis_Completed	تحلیل و شبیه‌سازی تغییرات توسط FunnelModel انجام شد
Report_Displayed	گزارش تغییرات و KPI نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها و تغییرات معماری
Requirement_Added	Analysis_Completed	تحلیل توسط FunnelModel
Analysis_Completed	Report_Displayed	نمایش داشبورد و KPI

از وضعیت	به وضعیت	رویداد / شرط
Report_Displayed	پایان چرخه و آماده برای تغییر بعدی Idle	

۶ Use Case

ارتباط	Use Case	Actor
Actor → System	ثبت تغییرات معماری و داده‌ها	User
Actor ← System	مشاهده Dashboard و KPI در CI/CD	User
Actor → System	مدیریت مدل Funnel و معیارهای تغییرات	System Admin
Actor ← System	مشاهده گزارش تغییرات و شاخص‌ها	System Admin

موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت گزارش KPI	ارسال/دریافت	User Smart Funnel System	
نمایش شاخص‌ها و وضعیت CI/CD	ارسال	Dashboard	Smart Funnel System
ارائه گزارش و شاخص‌ها	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها	Process 1	User
ذخیره داده‌ها	Process 1 User/Requirement	
مدیریت مدل تحلیل تغییرات	FunnelModel	Process 2
ثبت ریسک‌ها و شاخص‌های شبیه‌سازی RiskLog		Process 3
نمایش گزارش KPI	Dashboard	Process 4

DFD2

Entity / Sub-Process	ارتباط با	جریان داده
RiskLog	4.1	جمع‌آوری داده
4.1	4.2	Raw Data
4.2	4.3	KPI Report
4.3 Dashboard		نمایش

STD

Entity	ارتباط با	نقش
User Transitions		ایجاد و ثبت داده تغییرات
System Admin Transitions		مدیریت وضعیت‌ها و مدل
RiskLog	States	ثبت وضعیت و اطلاعات

Use Case Diagram

Actor	Use Case	جریان داده
User	ثبت داده تغییرات	ارسال داده‌ها
User	مشاهده KPI و Dashboard	دریافت گزارش‌ها
System Admin	Funnel	اصلاح و مدیریت داده‌ها و معیارهای تغییرات مدیریت مدل
System Admin	Dashboard	کنترل شاخص‌ها و گزارش‌ها

جمع‌بندی نهایی پروژه ۲۴ ✓

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب **AHP** و **ML** برای تحلیل و شبیه‌سازی تغییرات معماری در محیط **DevOps** و **CI/CD** ، همراه با ارزیابی ریسک و ارائه KPI

پروژه ۲۵ - مدل تصمیم‌یار برای انتخاب فناوری و ابزار در طراحی معماری کلان سازمانی

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department	User
نیازهای متغیر پروژه یا سیستم	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل فناوری و ابزارها	PK	model_id, name, version, description	FunnelModel
ثبت رویدادها و ریسک‌های مرتبط با انتخاب فناوری	PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسطه (M:N (User_Requirement

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده - دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Requirement	تخصیص به نیاز	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	Requirement FunnelModel		تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا	3	پر
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران برای انتخاب فناوری و ابزار مناسب در معماری کلان

جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

موجودیت

نقش / توضیح

وارد کردن داده‌های نیازها و محدودیت‌ها، مشاهده پیشنهاد فناوری User

Smart Funnel System تحلیل نیازها، ارزیابی فناوری و ابزارها، محاسبه ریسک

Dashboard نمایش شاخص‌ها، پیشنهاد فناوری، گزارش ریسک و KPI

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

Process	ورودی	خروجی
1. ثبت کاربران و نیازها	User, Requirement	داده‌های اولیه User/Requirement
2. مدیریت مدل تصمیم‌یار	FunnelModel	مدل آماده تحلیل فناوری
3. تحلیل فناوری و ریسک	User, Requirement, FunnelModel	RiskLog / داده‌های تحلیل و ریسک
4. تولید گزارش و شاخص‌ها	RiskLog	Dashboard / پیشنهاد فناوری و KPI

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog
→ Process 4 → Dashboard → User

DFD2 ۴.۳

جزئیات – Process 4 تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها (کارایی، هزینه، ریسک فناوری)	Raw Data	KPI Report	User, RiskLog
4.3 ارسال به داشبورد	KPI Report	Dashboard Display	Dashboard, User

۵ STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Requirement_Added	نیازها و محدودیت‌های فناوری ثبت شدند
Analysis_Completed	تحلیل فناوری و ابزارها توسط FunnelModel انجام شد
Report_Displayed	گزارش پیشنهاد فناوری و KPI نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها و محدودیت‌ها
Requirement_Added	Analysis_Completed	تحلیل توسط FunnelModel
Analysis_Completed	Report_Displayed	نمایش داشبورد و KPI

از وضعیت	به وضعیت	رویداد / شرط
Report_Displayed	پایان چرخه و آماده برای تحلیل بعدی Idle	

۶ Use Case

ارتباط	Use Case	Actor
Actor → System	ثبت نیازها و محدودیت‌های فناوری	User
Actor ← System	مشاهده Dashboard و پیشنهاد فناوری	User
Actor → System	مدیریت مدل Funnel و معیارهای فناوری	System Admin
Actor ← System	مشاهده گزارش و شاخص‌ها	System Admin

موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت پیشنهاد فناوری ارسال/دریافت			User Smart Funnel System
نمایش شاخص‌ها و پیشنهاد فناوری	ارسال	Dashboard	Smart Funnel System
ارائه پیشنهاد و KPI	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها	Process 1	User
ذخیره داده‌ها		Process 1 User/Requirement
مدیریت مدل تحلیل فناوری FunnelModel		Process 2
ثبت ریسک‌ها و تحلیل	RiskLog	Process 3
نمایش گزارش و KPI	Dashboard	Process 4

DFD2

جریان داده	ارتباط با	Entity / Sub-Process
جمع‌آوری داده 4.1		RiskLog
4.2 Raw Data	4.1	
4.3 KPI Report	4.2	
نمایش	4.3 Dashboard	

STD

نقش	ارتباط با	Entity
ایجاد و ثبت داده فناوری		User Transitions
مدیریت وضعیت‌ها و مدل		System Admin Transitions
ثبت وضعیت و اطلاعات	States	RiskLog

Use Case Diagram

جریان داده	Use Case	Actor
ارسال داده‌ها	ثبت داده فناوری	User
دریافت گزارش‌ها و پیشنهاد فناوری	مشاهده KPI و Dashboard	User
اصلاح و مدیریت مدل و معیارهای فناوری مدیریت مدل	Funnel	System Admin
کنترل شاخص‌ها و گزارش‌ها	مشاهده Dashboard	System Admin

جمع‌بندی نهایی پروژه ۲۵ ✓

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- User Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب **AHP** و **ML** برای انتخاب فناوری و ابزار مناسب در طراحی معماری کلان سازمانی، همراه با ارزیابی ریسک و KPI

پروژه ۲۶ - تحلیل ریسک ناسازگاری بین طراحی و استقرار در سیستم‌های توزیع‌شده

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department	User
نیازهای متغیر پروژه یا سیستم	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل ناسازگاری طراحی و استقرار	PK	model_id, name, version, description	FunnelModel
ثبت رویدادها و ریسک‌های تحلیل شده	PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسطه (M:N (User_Requirement)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده - دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement ، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Requirement	تخصیص به نیاز	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	Requirement FunnelModel		تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا	3	پر
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران و تحلیل ناسازگاری‌ها

جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

موجودیت

نقش / توضیح

وارد کردن داده‌های طراحی و مستندات استقرار، مشاهده گزارش ناسازگاری‌ها User

Smart Funnel System تحلیل ناسازگاری طراحی و استقرار، محاسبه ریسک و پیشنهاد اصلاح

Dashboard نمایش شاخص‌ها، گزارش ریسک و پیشنهادات اصلاحی

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

Process

ورودی

خروجی

1. ثبت کاربران و نیازها

User, Requirement

داده‌های اولیه User/Requirement

2. مدیریت مدل تصمیم‌یار

FunnelModel

مدل آماده تحلیل ناسازگاری

خروجی	ورودی	Process
RiskLog / داده‌های ریسک و ناسازگاری	User, Requirement, FunnelModel	3. تحلیل ناسازگاری و ریسک
Dashboard / گزارش و پیشنهاد اصلاح	RiskLog	4. تولید گزارش و شاخص‌ها

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog → Process 4 → Dashboard → User

۴.۳ DFD2

جزئیات – Process 4 تولید گزارش و شاخص‌ها

Related Entities	Output	Input	Sub-Process
RiskLog	Raw Data	RiskLog	4.1 جمع‌آوری داده‌ها
User, RiskLog	KPI Report	Raw Data	4.2 محاسبه شاخص‌ها (شدت ناسازگاری، احتمال خطا، تاثیر بر عملکرد)
Dashboard, User	Dashboard Display	KPI Report	4.3 ارسال به داشبورد

۵ STD – State Transition Diagram

توضیح	States
سیستم منتظر داده است	Idle
کاربران ثبت شدند	User_Registered
نیازها و مستندات طراحی/استقرار ثبت شدند	Requirement_Added
تحلیل ناسازگاری‌ها توسط FunnelModel انجام شد	Analysis_Completed
گزارش ریسک و پیشنهاد اصلاح نمایش داده شد	Report_Displayed

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها و مستندات طراحی/استقرار
Requirement_Added	Analysis_Completed	تحلیل توسط FunnelModel
Analysis_Completed	Report_Displayed	نمایش داشبورد و KPI
Report_Displayed	Idle	پایان چرخه و آماده برای تحلیل بعدی

۶. Use Case

Actor	Use Case	ارتباط
User	ثبت اطلاعات طراحی و استقرار	Actor → System
User	مشاهده Dashboard و گزارش ناسازگاری‌ها	Actor ← System
System Admin	مدیریت مدل Funnel و معیارهای ریسک	Actor → System
System Admin	مشاهده گزارش KPI	Actor ← System

موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت گزارش ناسازگاری ارسال/دریافت			User Smart Funnel System
نمایش شاخص‌ها و گزارش اصلاح	ارسال	Dashboard	Smart Funnel System
ارائه پیشنهاد اصلاح و KPI	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها	Process 1	User

Entity / Process	ارتباط با	جریان داده
Process 1	User/Requirement	ذخیره داده‌ها
Process 2	FunnelModel	مدیریت مدل تحلیل ناسازگاری
Process 3	RiskLog	ثبت ریسک‌ها و ناسازگاری‌ها
Process 4	Dashboard	نمایش گزارش و KPI

DFD2

Entity / Sub-Process	ارتباط با	جریان داده
RiskLog	4.1	جمع‌آوری داده
4.1	4.2	Raw Data
4.2	4.3	KPI Report
4.3	Dashboard	نمایش

STD

Entity	ارتباط با	نقش
User Transitions		ایجاد و ثبت داده
System Admin Transitions		مدیریت وضعیت‌ها و مدل
RiskLog	States	ثبت وضعیت و اطلاعات

Use Case Diagram

Actor	Use Case	جریان داده
User	ثبت داده طراحی/استقرار	ارسال داده‌ها
User	مشاهده Dashboard	دریافت گزارش‌ها و پیشنهاد اصلاح
System Admin	Funnel	اصلاح و مدیریت مدل و معیارهای ریسک مدیریت مدل
System Admin	مشاهده Dashboard	کنترل شاخص‌ها و گزارش‌ها

جمع‌بندی نهایی پروژه ۲۶ 

• DFD0: نمای کلی و جعبه سیاه

- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها و گزارش‌دهی
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب AHP و ML برای شناسایی ناسازگاری بین طراحی و استقرار در سیستم‌های توزیع‌شده و پیشنهاد اقدامات اصلاحی بر اساس ریسک و KPI

پروژه ۲۷ – مدل قیف مستندسازی معماری با تولید خودکار نمودار از کد منبع

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department	User
نیازهای پروژه با سیستم برای مستندسازی	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل نیازها و تولید نمودار	PK	model_id, name, version, description	FunnelModel
ثبت رویدادها، ریسک‌ها و ناسازگاری‌ها	PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسط (M:N) (User_Requirement)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری و تولید نمودار - اشتباه شود		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement ، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Requirement	تخصیص به نیاز	M:N	بالقوه	زرد	متوسط	2	نقطه‌چین
2	Requirement FunnelModel		تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا	3	پر
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه‌چین

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران و فرآیند مستندسازی

جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

موجودیت

نقش / توضیح

User وارد کردن کد منبع، ثبت نیازها و دریافت گزارش/نمودار

Smart Funnel System تحلیل نیازها، استخراج معماری، تولید خودکار نمودار و مستندات

Dashboard نمایش نمودارها، گزارش ریسک و KPI مستندسازی

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

خروجی	ورودی	Process
داده‌های اولیه User/Requirement	User, Requirement	1. ثبت کاربران و نیازها
مدل آماده تحلیل و استخراج نمودار	FunnelModel	2. مدیریت مدل تصمیم‌یار
RiskLog / نمودار و مستندات معماری	User, Requirement, FunnelModel	3. تحلیل ریسک و تولید نمودار خودکار
Dashboard / گزارش KPI	RiskLog	4. تولید گزارش و شاخص‌ها

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog
→ Process 4 → Dashboard → User نمودار+

۴.۳ DFD2

جزئیات – Process 4 تولید گزارش و نمودارها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها و تحلیل کیفیت مستندات	Raw Data	KPI Report	User, RiskLog
4.3 تولید نمودار خودکار معماری	Raw Data	Architecture Diagram	FunnelModel, Dashboard
4.4 ارسال به داشبورد	KPI Report + Diagram	Dashboard Display	Dashboard, User

۵ . STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده و کد منبع است
User_Registered	کاربران ثبت شدند

States	توضیح
Requirement_Added	نیازها و کد منبع ثبت شدند
Analysis_Completed	تحلیل توسط FunnelModel انجام شد، نمودار تولید شد
Report_Displayed	نمودار و گزارش نمایش داده شد

Transitions:		
از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها و کد منبع
Requirement_Added	Analysis_Completed	تحلیل و تولید نمودار خودکار
Analysis_Completed	Report_Displayed	نمایش داشبورد و نمودار
Report_Displayed	Idle	پایان چرخه و آماده برای تحلیل بعدی

۶. Use Case

Actor	Use Case	ارتباط
User	Actor → System ثبت اطلاعات و کد منبع	
User	Actor ← System مشاهده Dashboard و نمودارهای معماری	
System Admin	Actor → System مدیریت مدل Funnel و معیارهای تحلیل	
System Admin	Actor ← System مشاهده نمودار و گزارش KPI	

موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت نمودار و گزارش ارسال/دریافت			User Smart Funnel System

جریان داده	نوع ارتباط	ارتباط با	Entity
نمایش نمودار و KPI مستندسازی	ارسال	Dashboard	Smart Funnel System
ارائه نمودار و گزارش	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها و کد منبع	Process 1	User
ذخیره داده‌ها	Process 1 User/Requirement	
مدیریت مدل تحلیل و تولید نمودار	FunnelModel	Process 2
ثبت ریسک‌ها و ناسازگاری‌ها	RiskLog	Process 3
نمایش نمودار و KPI	Dashboard	Process 4

DFD2

جریان داده	ارتباط با Entity / Sub-Process	
جمع‌آوری داده	4.1	RiskLog
Raw Data	4.2	4.1
4.3 Diagram + KPI Report	4.3	4.2
Dashboard Display	4.4	4.3

STD

نقش	ارتباط با	Entity
ایجاد و ثبت داده	User Transitions	
مدیریت وضعیت‌ها و مدل	System Admin Transitions	
ثبت وضعیت و اطلاعات	States	RiskLog

Use Case Diagram

جریان داده	Use Case	Actor
ارسال داده‌ها	ثبت داده و کد منبع	User
دریافت گزارش و نمودار	مشاهده Dashboard و نمودارها	User

Actor	Use Case	جریان داده
System Admin	Funnel	اصلاح و مدیریت مدل و معیارهای ریسک مدیریت مدل
System Admin	Dashboard	کنترل شاخص‌ها و نمودارهای معماری مشاهده

✓ جمع‌بندی نهایی پروژه ۲۷

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها، KPI و تولید نمودار
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب **AHP** و **ML** برای مستندسازی معماری از کد منبع، تولید خودکار نمودار و گزارش **KPI** بر اساس معیارهای کیفی، ریسک و نیاز پروژه

پروژه ۲۸ – طراحی سیستم هوشمند ارزیابی سبک‌های معماری

۱. جدول پایه موجودیت‌ها (Entities)

Entity	Attributes	Type Key	توضیح
User	user_id, username, role, department	INT, VARCHAR, VARCHAR, VARCHAR PK	اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)
Requirement	req_id, name, type, priority, status	INT, VARCHAR, VARCHAR, VARCHAR, VARCHAR PK	نیازهای پروژه یا سیستم برای ارزیابی سبک معماری
FunnelModel	model_id, name, version, description	INT, VARCHAR, VARCHAR, VARCHAR PK	مدل تصمیم‌یار هوشمند برای تحلیل نیازها و پیشنهاد سبک معماری بهینه
RiskLog	log_id, req_id, model_id, severity, date, notes	INT, INT, INT, VARCHAR, DATE, VARCHAR PK	ثبت رویدادها، ریسک‌ها و ناسازگاری‌ها

۲. جدول واسطه (M:N (User_Requirement

رиск (نام و توضيح)	كليد	نوع داده	توضيح	ستون
رکورد تکراری یا گمشده – دسترسى نادرست کاربر	PK	INT	شناسه يک‌تا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحليل منطق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است – تصميم‌گیری و ارزیابی سبک اشتباه شود		VARCHAR	نقش کاربر در تحليل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement ، ريسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ريسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ريسک

نوع خط	ضخامت خط	سطح هوشیاری	رنگ ريسک	نوع ريسک	نوع رابطه	نام رابطه	Entity هدف	Entity منبع	#
نقطه‌چین 2	متوسط	زرد	بالقوه	M:N	تخصیص به نیاز		User Requirement		1
پر 3	بالا	قرمز	بالفعل	1:N	تحليل توسط مدل		Requirement FunnelModel		2
پر 3	بالا	قرمز	بالفعل	1:N	ثبت ريسک		RiskLog	User	3
نقطه‌چین 2	متوسط	زرد	بالقوه	1:N	مرتبط با نیاز		RiskLog	Requirement	4

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران و ارزیابی سبک معماری

جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

نقش / توضیح

موجودیت

وارد کردن نیازها و معیارهای پروژه، دریافت توصیه سبک معماری User

تحلیل نیازها، ارزیابی سبک‌های معماری و پیشنهاد سبک بهینه Smart Funnel System

نمایش سبک‌های پیشنهادی، نمودارها و گزارش KPI Dashboard

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

Process	ورودی	خروجی
1. ثبت کاربران و نیازها	User, Requirement	داده‌های اولیه User/Requirement
2. مدیریت مدل تصمیم‌یار	FunnelModel	مدل آماده تحلیل و ارزیابی سبک
3. تحلیل ریسک و ارزیابی سبک	User, Requirement, FunnelModel	RiskLog / سبک پیشنهادی و داده‌های ریسک
4. تولید گزارش و شاخص‌ها	RiskLog	Dashboard / گزارش و نمودار سبک معماری

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog
 → Process 4 → Dashboard → User + سبک پیشنهادی

۴.۳ DFD2

جزئیات – Process 4 تولید گزارش و نمودارهای سبک معماری

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها و ارزیابی سبک‌ها	Raw Data	KPI Report	User, RiskLog
4.3 تولید نمودار سبک معماری	Raw Data	Architecture Style Diagram	FunnelModel, Dashboard
4.4 ارسال به داشبورد	KPI Report + Diagram	Dashboard Display	Dashboard, User

۵. STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Requirement_Added	نیازها ثبت شدند
Analysis_Completed	تحلیل توسط FunnelModel انجام شد، سبک بهینه انتخاب شد
Report_Displayed	نمودار و گزارش نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها و معیارهای پروژه
Requirement_Added	Analysis_Completed	تحلیل و انتخاب سبک معماری
Analysis_Completed	Report_Displayed	نمایش داشبورد و نمودار سبک معماری
Report_Displayed	Idle	پایان چرخه و آماده برای تحلیل بعدی

۶. Use Case

Actor	Use Case	ارتباط
User	ثبت اطلاعات و معیارهای پروژه	Actor → System
User	مشاهده Dashboard و سبک پیشنهادی	Actor ← System
System Admin	مدیریت مدل Funnel و معیارهای تحلیل	Actor → System
System Admin	مشاهده نمودار سبک و گزارش KPI	Actor ← System

موجودیت‌ها / ارتباطها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت سبک پیشنهادی ارسال/دریافت User Smart Funnel System			
نمایش سبک معماری، نمودار و KPI ارسال	Dashboard	Smart Funnel System	
ارائه سبک پیشنهادی و گزارش	User	Dashboard	

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها و نیازها	Process 1	User
ذخیره داده‌ها	Process 1 User/Requirement	
مدیریت مدل تحلیل و ارزیابی سبک FunnelModel	Process 2	
ثبت ریسک‌ها و ارزیابی سبک RiskLog	Process 3	
نمایش نمودار و KPI سبک معماری Dashboard	Process 4	

DFD2

جریان داده	ارتباط با	Entity / Sub-Process
جمع‌آوری داده	4.1	RiskLog
Raw Data	4.2	4.1
4.3 Diagram + KPI Report	4.3	4.2
Dashboard Display	4.4	4.3

STD

نقش	ارتباط با	Entity
ایجاد و ثبت داده	User Transitions	
مدیریت وضعیت‌ها و مدل	System Admin Transitions	
ثبت وضعیت و اطلاعات States	RiskLog	

Use Case Diagram

Actor	Use Case	جریان داده
User	ثبت داده و معیارها	ارسال داده‌ها
User	مشاهده Dashboard و سبک پیشنهادی	دریافت گزارش و نمودار
System Admin	مدیریت مدل Funnel	اصلاح و مدیریت مدل و معیارها
System Admin	Dashboard مشاهده معماری	کنترل شاخص‌ها و نمودار سبک معماری

✓ جمع‌بندی نهایی پروژه ۲۸

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها، KPI و نمودار سبک معماری
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب **AHP** و **ML** برای ارزیابی سبک‌های معماری (Layered, Event-driven, MVC...) و پیشنهاد سبک بهینه بر اساس معیارهای کیفی، هزینه و ریسک

پروژه ۲۹ – مدل قیف امنیت معماری برای پیش‌بینی نقاط آسیب‌پذیر سیستم

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر امنیت، مدیر)	PK	user_id, username, role, department INT, VARCHAR, VARCHAR, VARCHAR	User
نیازها و الزامات امنیتی پروژه یا سیستم	PK	req_id, name, type, priority, status INT, VARCHAR, VARCHAR, VARCHAR, VARCHAR	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل نیازها و پیش‌بینی نقاط آسیب‌پذیر	PK	model_id, name, version, description INT, VARCHAR, VARCHAR, VARCHAR	FunnelModel

توضیح	Type Key	Attributes	Entity
ثبت رویدادها، ریسک‌ها و نقاط آسیب‌پذیر پیش‌بینی شده	INT, INT, INT, VARCHAR, DATE, PK VARCHAR	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسطه (User_Requirement) M:N

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری و تحلیل امنیتی – اشتباه شود		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

نوع خط	ضخامت خط	سطح هوشیاری	رنگ ریسک	نوع ریسک	نوع رابطه	نام رابطه	Entity هدف	Entity منبع	#
نقطه چین 2		متوسط	زرد	بالحقوه	M:N	تخصیص به نیاز	User Requirement		1
پر 3		بالا	قرمز	بالفعل	1:N	تحلیل توسط مدل	Requirement FunnelModel		2
پر 3		بالا	قرمز	بالفعل	1:N	ثبت ریسک	RiskLog	User	3
نقطه چین 2		متوسط	زرد	بالحقوه	1:N	مرتبط با نیاز	RiskLog	Requirement	4

۴. نمودارهای فرآیند و رفتار سیستم

DFD0 ۴.۱

هدف: تعامل کلی سیستم با کاربران و تحلیل نقاط آسیب‌پذیر

جریان داده:

[User] → [Security Funnel System] → [Dashboard / Reports] → [User]

نقش / توضیح	موجودیت
وارد کردن نیازها، معیارهای امنیتی و الزامات پروژه	User
تحلیل ریسک امنیتی و پیش‌بینی نقاط آسیب‌پذیر	Security Funnel System
نمایش شاخص‌ها، نمودارهای ریسک و گزارش پیش‌بینی‌ها	Dashboard

DFD1 ۴.۲

فرآیندها و جریان داده‌ها:

خروجی	ورودی	Process
داده‌های اولیه User/Requirement	User, Requirement	1. ثبت کاربران و نیازها
مدل آماده تحلیل و پیش‌بینی	FunnelModel	2. مدیریت مدل تصمیم‌یار امنیتی
RiskLog / نقاط آسیب‌پذیر شناسایی شده	User, Requirement, FunnelModel	3. تحلیل ریسک و پیش‌بینی نقاط آسیب‌پذیر
Dashboard / گزارش و نمودار نقاط آسیب‌پذیر	RiskLog	4. تولید گزارش و شاخص‌ها

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog
→ Process 4 → Dashboard → User

DFD2 ۴.۳

جزئیات – Process 4 تولید گزارش و شاخص‌های امنیتی

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها و اولویت‌بندی نقاط آسیب‌پذیر	Raw Data	KPI Report	User, RiskLog

Sub-Process	Input	Output	Related Entities
4.3 تولید نمودار نقاط آسیب‌پذیر	Raw Data	Vulnerability Diagram	FunnelModel, Dashboard
4.4 ارسال به داشبورد	KPI Report + Diagram	Dashboard Display	Dashboard, User

۵ STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده است
User_Registered	کاربران ثبت شدند
Requirement_Added	نیازها و معیارهای امنیتی ثبت شدند
Analysis_Completed	تحلیل امنیتی و پیش‌بینی نقاط آسیب‌پذیر توسط FunnelModel انجام شد
Report_Displayed	نمودار و گزارش پیش‌بینی نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها و الزامات امنیتی
Requirement_Added	Analysis_Completed	تحلیل و پیش‌بینی نقاط آسیب‌پذیر
Analysis_Completed	Report_Displayed	نمایش داشبورد و نمودار نقاط آسیب‌پذیر
Report_Displayed	Idle	پایان چرخه و آماده برای تحلیل بعدی

۶ Use Case

Actor	Use Case	ارتباط
User	ثبت اطلاعات و معیارهای امنیتی پروژه	Actor → System
User	مشاهده Dashboard و نمودار نقاط آسیب‌پذیر	Actor ← System

Actor	Use Case	ارتباط
System Admin	Actor → System مدیریت مدل Funnel و معیارهای تحلیل امنیتی	
System Admin	Actor ← System مشاهده نمودار امنیتی و گزارش KPI	
موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel		

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و دریافت گزارش نقاط آسیب‌پذیر	ارسال/دریافت	Security Funnel System	User
نمایش نمودار و شاخص‌های امنیتی	ارسال	Dashboard	Security Funnel System
ارائه نمودار نقاط آسیب‌پذیر و گزارش KPI	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها و الزامات امنیتی	Process 1	User
ذخیره داده‌ها	Process 1 User/Requirement	
مدیریت مدل تحلیل و پیش‌بینی نقاط آسیب‌پذیر	FunnelModel	Process 2
ثبت ریسک‌ها و نقاط آسیب‌پذیر	RiskLog	Process 3
نمایش نمودار و KPI امنیتی	Dashboard	Process 4

DFD2

جریان داده	ارتباط با	Entity / Sub-Process
جمع‌آوری داده	4.1	RiskLog
Raw Data	4.1 4.2	

جریان داده ارتباط با Entity / Sub-Process

4.2 4.3 Diagram + KPI Report

4.3 4.4 Dashboard Display

STD

نقش	ارتباط با	Entity
ایجاد و ثبت داده		User Transitions
مدیریت وضعیت‌ها و مدل		System Admin Transitions
ثبت وضعیت و اطلاعات	States	RiskLog

Use Case Diagram

جریان داده	Use Case	Actor
ارسال داده‌ها	ثبت داده و معیارهای امنیتی	User
دریافت گزارش و نمودار	مشاهده Dashboard و نمودار نقاط آسیب‌پذیر	User
اصلاح و مدیریت مدل و معیارها	مدیریت مدل Funnel	System Admin
کنترل شاخص‌ها و نمودار نقاط آسیب‌پذیر	مشاهده Dashboard	System Admin

جمع‌بندی نهایی پروژه ۲۹

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها، KPI و نمودار نقاط آسیب‌پذیر
- STD: وضعیت‌ها و انتقال‌ها
- Use Case: تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب AHP و ML برای پیش‌بینی نقاط آسیب‌پذیر معماری سیستم و پیشنهاد راهکار کاهش ریسک امنیتی

پروژه ۳۰ – پروژه تلفیقی: طراحی، تحلیل و ارزیابی کامل یک سیستم هوشمند با معماری قیفی و پرسشنامه ۲۵ سؤالی

۱. جدول پایه موجودیت‌ها (Entities)

توضیح	Type Key	Attributes	Entity
اطلاعات کاربران (معمار، تحلیل‌گر، مدیر)	PK	user_id, username, role, department	User
نیازها و الزامات پروژه یا سیستم	PK	req_id, name, type, priority, status	Requirement
مدل تصمیم‌یار هوشمند برای تحلیل نیازها و ارائه توصیه‌ها	PK	model_id, name, version, description	FunnelModel
ثبت رویدادها، ریسک‌ها و تحلیل‌های تصمیم‌یار	PK	log_id, req_id, model_id, severity, date, notes	RiskLog

۲. جدول واسطه (M:N (User_Requirement)

ریسک (نام و توضیح)	کلید	نوع داده	توضیح	ستون
رکورد تکراری یا گمشده – دسترسی نادرست کاربر	PK	INT	شناسه یکتا هر رکورد	ur_id
داده‌ها با کاربر واقعی مطابقت ندارد	FK → User(user_id)	INT	ارجاع به جدول User	user_id
نیاز واقعی با تحلیل منطبق نیست	FK → Requirement(req_id)	INT	ارجاع به جدول Requirement	req_id
ثبت نقش اشتباه ممکن است تصمیم‌گیری را غلط کند		VARCHAR	نقش کاربر در تحلیل نیاز (Owner/Reviewer)	role

ویژگی‌ها: مدیریت M:N بین User و Requirement، ریسک روی صحت داده‌ها و نقش‌ها، سه بخش در Draw.io با رنگ‌بندی ریسک (سبز/زرد/قرمز)

۳. جدول روابط بین موجودیت‌ها با ویژگی‌های ریسک

#	Entity منبع	Entity هدف	نام رابطه	نوع رابطه	نوع ریسک	رنگ ریسک	سطح هوشیاری	ضخامت خط	نوع خط
1	User	Requirement	تخصیص به نیاز	M:N	بالقوه	زرد	متوسط	2	نقطه چین
2	Requirement FunnelModel		تحلیل توسط مدل	1:N	بالفعل	قرمز	بالا	3	پر
3	User	RiskLog	ثبت ریسک	1:N	بالفعل	قرمز	بالا	3	پر
4	Requirement	RiskLog	مرتبط با نیاز	1:N	بالقوه	زرد	متوسط	2	نقطه چین

۴. نمودارهای فرآیند و رفتار سیستم

۴.۱ DFD0

هدف: تعامل کلی سیستم با کاربران و ارائه تحلیل تصمیم‌یار

جریان داده:

[User] → [Smart Funnel System] → [Dashboard / Reports] → [User]

موجودیت

نقش / توضیح

User وارد کردن نیازها، پاسخ به پرسشنامه ۲۵ سوالی، مشاهده گزارش‌ها

تحلیل داده‌ها با مدل تصمیم‌یار AHP + ML و پیش‌بینی بهترین سبک معماری Smart Funnel System

Dashboard نمایش شاخص‌ها، نمودار تصمیم‌گیری و توصیه‌های معماری

۴.۲ DFD1

فرآیندها و جریان داده‌ها:

Process	ورودی	خروجی
1. ثبت کاربران و نیازها	User, Requirement	داده‌های اولیه User/Requirement
2. مدیریت مدل تصمیم‌یار	FunnelModel	مدل آماده تحلیل و پیشنهاد معماری
3. تحلیل ریسک و امتیازدهی به سبک‌ها	User, Requirement, FunnelModel	RiskLog / داده‌های ریسک و سبک پیشنهادی
4. تولید گزارش و شاخص‌ها	RiskLog	Dashboard / گزارش و نمودار تصمیم‌گیری

جریان داده:

User → Process 1 → Requirement/User → Process 2 → FunnelModel → Process 3 → RiskLog
→ Process 4 → Dashboard → User

DFD2 ۴.۳

جزئیات – Process 4 تولید گزارش و شاخص‌ها

Sub-Process	Input	Output	Related Entities
4.1 جمع‌آوری داده‌ها	RiskLog	Raw Data	RiskLog
4.2 محاسبه شاخص‌ها و امتیازدهی سبک‌ها	Raw Data	KPI Report	User, RiskLog
4.3 تولید نمودار تصمیم‌گیری	KPI Report	Architecture Recommendation Diagram	FunnelModel, Dashboard
4.4 ارسال به داشبورد	Diagram + KPI Report	Dashboard Display	Dashboard, User

۵ STD – State Transition Diagram

States	توضیح
Idle	سیستم منتظر داده‌ها و پاسخ پرسشنامه است
User_Registered	کاربران ثبت شدند
Requirement_Added	نیازها و پرسشنامه‌های ۲۵ سوالی ثبت شدند
Analysis_Completed	تحلیل توسط FunnelModel و امتیازدهی سبک‌ها انجام شد
Report_Displayed	نمودار، شاخص و پیشنهاد معماری نمایش داده شد

Transitions:

از وضعیت	به وضعیت	رویداد / شرط
Idle	User_Registered	ثبت کاربر
User_Registered	Requirement_Added	ثبت نیازها و پاسخ پرسشنامه
Requirement_Added	Analysis_Completed	تحلیل و امتیازدهی سبک‌ها

از وضعیت	به وضعیت	رویداد / شرط
Analysis_Completed	Report_Displayed	نمایش داشبورد و نمودار تصمیمگیری
Report_Displayed	Idle	پایان چرخه و آماده برای تحلیل بعدی

۶. Use Case

Actor	Use Case	ارتباط
User	Actor → System	ثبت اطلاعات و پاسخ پرسشنامه ۲۵ سوالی
User	Actor ← System	مشاهده Dashboard و نمودار سبک پیشنهادی
System Admin	Actor → System	مدیریت مدل Funnel و معیارهای تحلیل
System Admin	Actor ← System	مشاهده Dashboard و KPI های تصمیمگیری

موجودیت‌ها / ارتباط‌ها : User, System Admin, Dashboard, Requirement, RiskLog, FunnelModel

۷. ارتباط موجودیت‌ها و جریان داده‌ها

DFD0

جریان داده	نوع ارتباط	ارتباط با	Entity
ورود داده‌ها و پرسشنامه و دریافت گزارش	ارسال/دریافت	Smart Funnel System	User
نمایش نمودار سبک پیشنهادی و KPI	ارسال	Dashboard	Smart Funnel System
ارائه نمودار و توصیه‌های معماری	نمایش	User	Dashboard

DFD1

جریان داده	ارتباط با	Entity / Process
ارسال داده‌ها و پاسخ پرسشنامه	Process 1	User
ذخیره داده‌ها	Process 1 User/Requirement	
مدیریت مدل تحلیل و سبک پیشنهادی FunnelModel	Process 2	

Entity / Process	ارتباط با	جریان داده
Process 3	RiskLog	ثبت ریسک‌ها و امتیازدهی سبک‌ها
Process 4	Dashboard	نمایش نمودار و KPI تصمیم‌گیری

DFD2

Entity / Sub-Process	ارتباط با	جریان داده
RiskLog	4.1	جمع‌آوری داده
4.1	4.2	Raw Data
4.2	4.3	Diagram + KPI Report
4.3	4.4	Dashboard Display

STD

Entity	ارتباط با	نقش
User Transitions		ایجاد و ثبت داده
System Admin Transitions		مدیریت وضعیت‌ها و مدل
RiskLog	States	ثبت وضعیت و اطلاعات

Use Case Diagram

Actor	Use Case	جریان داده
User	ثبت داده و پرسشنامه ۲۵ سوالی	ارسال داده‌ها
User	مشاهده Dashboard و سبک پیشنهادی	دریافت گزارش و نمودار
System Admin	مدیریت مدل Funnel	اصلاح و مدیریت مدل و معیارها
System Admin	Dashboard تصمیم‌گیری مشاهده	کنترل شاخص‌ها و نمودار تصمیم‌گیری

جمع‌بندی نهایی پروژه ۳۰ 

- DFD0: نمای کلی و جعبه سیاه
- DFD1: فرآیندهای اصلی + جریان داده‌ها
- DFD2: جزئیات دقیق فرآیندها، KPI و نمودار سبک پیشنهادی
- STD: وضعیت‌ها و انتقال‌ها

- Use Case:تعامل Actor ها و سیستم
- جدول واسط: User_Requirement: فیلدها و ریسک‌های مخصوص خود، سه بخش در Draw.io
- پرسشنامه ۲۵ سؤالی: جمع‌آوری داده‌ها از کاربران برای امتیازدهی سبک‌ها
- هدف نهایی: ایجاد سیستم تصمیم‌یار هوشمند با ترکیب **AHP** و **ML** برای پیشنهاد بهترین سبک معماری بر اساس معیارهای کیفی، هزینه و ریسک