

SWE 4663
Software Project Management
Term Project Notes (Not a deliverable)

4/8/2025

✓ **Definite Goal / Project Objective**

You are building a **Project Management System** (desktop application) that allows users to:

- Track software development projects.
- Log and report hours spent across phases (e.g., planning, coding, testing).
- Manage project requirements and risks.
- Control user access based on roles.
- Generate real-time dashboards, graphs, and downloadable reports.

Primary goals:

- Enable efficient project tracking.
- Improve team collaboration.
- Enhance transparency in effort estimation.
- Support data-driven decision-making.

A. A General section that allows the input of the following information:

1. A high-level description of the software project
2. The owner or the project manager's name
3. A list of project team members (which may vary as the project progresses)
4. A list of risks and risk status

B. Project Requirements

1. Have a feature that allows the user to enter a list of functional and non-functional requirements for the software project

C. Project Effort Monitoring and Tracking

1. Have a feature that allows the user to enter (at a daily or weekly basis) the effort in the number of person hours expended on Requirements Analysis, Designing, Coding, Testing, Project Management (by each of the earlier entered requirements)
2. Have a feature that allows the user to view the total expended hours by requirements analysis, designing, coding, testing, project management (by each of the requirement entered).

Tech Stack

Area	Tools / Languages
Programming Language	C#
Framework	.NET 8.0
UI Type	Avalonia UI
Database	sqlite
Version Control	GitHub

Team Roles & Responsibilities

Member	Role	Responsibilities
Kahmin Keller	Project Manager	Oversees project execution and deadlines
Kevin Syhavong	Lead Developer	Manages core architecture and development
RJ Straiton	UI/UX Designer	Designs user interface and experience
Patrick Cox	QA Engineer	Tests system functionality and performance
Jared Louissaint	Documentation & Research	Prepares documentation and conducts feasibility studies

Major Deliverables

- Fully functional software system
 - User and technical documentation
 - Test reports
 - Final project presentation
-



Agile-Based Schedule

Phase	Duration	Description
Requirements Analysis	1 week	Define system requirements
Design Phase	2 weeks	System architecture and UI design
Development	4 weeks	Build core features
Testing & Debugging	2 weeks	QA and bug fixing
Final Documentation	1 week	Manuals, reports, and final polish

Risks & Mitigation

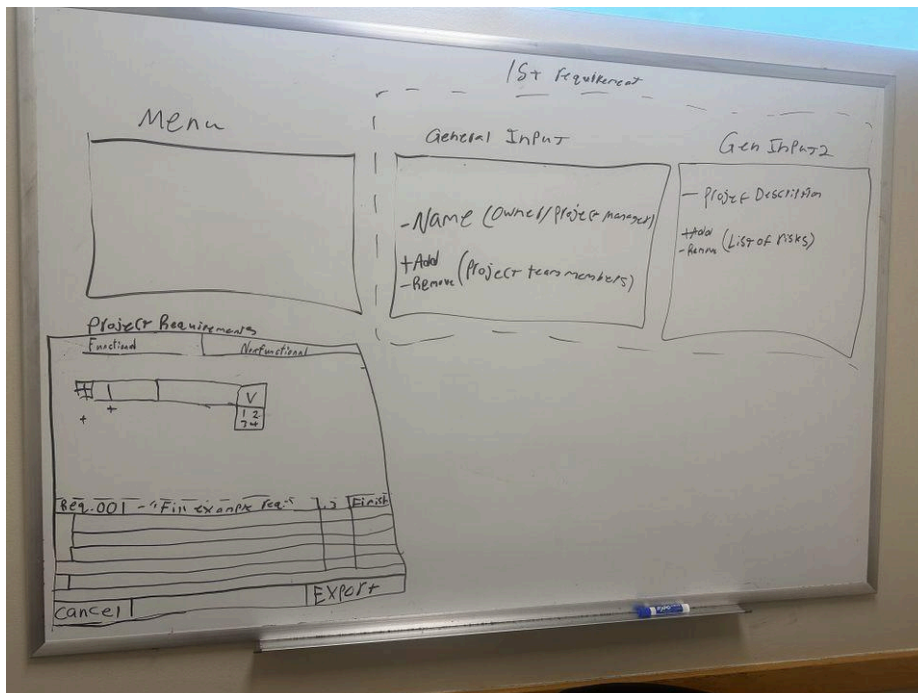
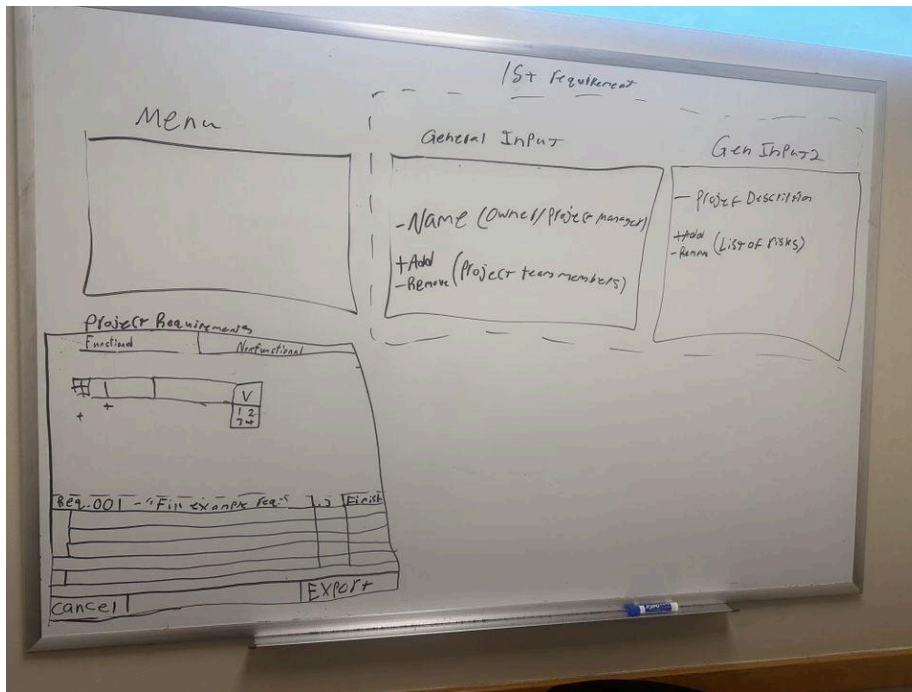
Risk	Impact	Mitigation
Scope Creep	High	Lock down scope early; review frequently
Team Coordination Issues	Medium	Use structured comms (e.g., weekly meetings)
Time Constraints	High	Prioritize + allocate buffer time
Technical Challenges	High	Feasibility studies; research ahead

Things to Discuss in the Meeting

1. **Reconfirm everyone's responsibilities** – especially for coding vs documentation.
2. **Decide on your source control and workflow strategy** (e.g., branching strategy on GitHub).
3. **Start breaking up development into actual tasks/stories** for sprints.
4. **Define early prototypes or mockups** for the UI.
5. **Establish meeting cadence** (e.g., weekly stand-ups).
6. **Set deadlines for key deliverables** (milestone planning).

7. **Clarify testing plan** – who's writing unit tests, when to do integration testing.
 8. **Talk about documentation ownership** – who updates what and when.
-

4/9/2025





Team Roles & Responsibilities

Member	Role	Responsibilities
Kahmin Keller	Project Manager	Oversees project execution and deadlines
Kevin Syhavong	Lead Developer	Manages core architecture and development
RJ Straiton	UI/UX Designer	Designs user interface and experience
Patrick Cox	QA Engineer	Tests system functionality and performance
Jared Louissaint	Documentation & Research	Prepares documentation and conducts feasibility studies

Database = SQL (MSFT Access)

Need to get homepage up an running (axml)

Cap requirements (parent and child) @ 999

Side note: Maybe get gitKraken

Log in

Organization ID: mainuser

Password: password

<https://www.w3schools.com/>

Definites from this meeting

Kevin & Patrick = work on viewModel and MainWindowView model (familiarization and getting something going) For next meeting: work on getting screen format to 2 text inputs and a button

Khamin = back and front end (support)

RJ = backend and database

Jared = whatever he can

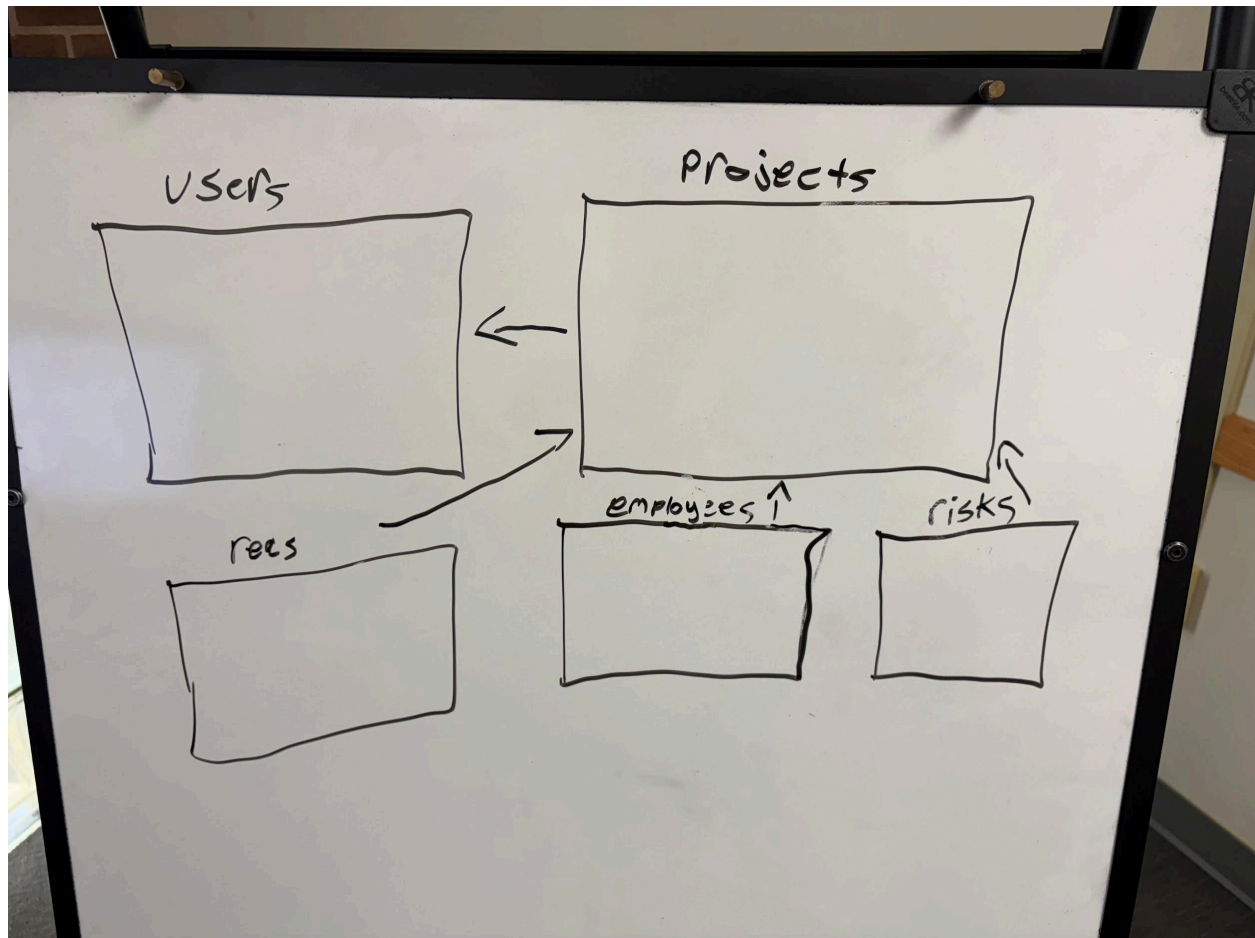
PK	ID(s)	Name(s)	Desc(s)	

10/01
1.1.1



1 1.1 1.1.1
2 2.1 2.1.1
3 3.1 3.1.1

20 20.1 20.1.1



4/23/2025

What's done already:

- Main screen and log in
- Existing projects (with scroll wheel, create new project button, and delete project button)

What needs to be done:

- Project requirements page (functional and nonfunctional): specific items required for a project (example: functional = user will be able to log in and log out, non-functional = system shall respond to user within specific time frame within log in) **Patrick**

Project title	Project id	requirementl D	Requirement Name	Description	Status	priority

- Risk page: list of risk associated with a specific project (scope creep, time management) **Jared**

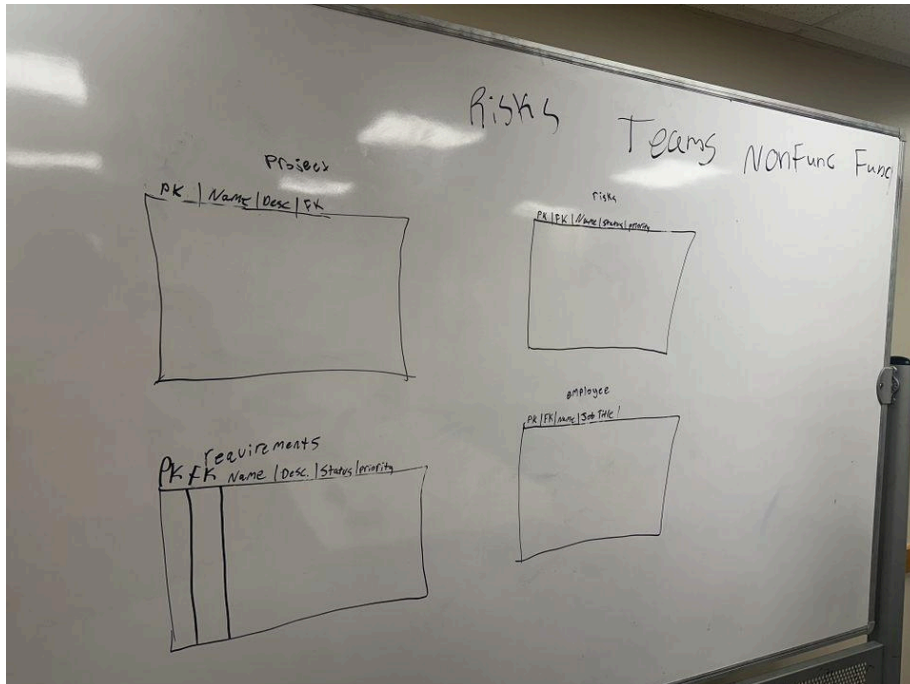
Project title	Project id	Risk associated			
---------------	------------	-----------------	--	--	--

- Employee page: Basic info about specific users (example: ID number) **Khamin**

Employee name	Employee ID				
---------------	-------------	--	--	--	--

- Productivity tracking (example: Hours total for whole group) **Kevin**

Project	Phase name (analysis, design, etc)	Weekly hrs	Tot hrs	Description (maybe)	
---------	--	------------	---------	------------------------	--



All other images/page illustrations below

View model = how other pages will interact with other pages

View = page itself

IS+ Requirement

Menu

- Name (Owner/Project Manager)
- + Add (Project team members)
- Remove (Project team members)

— Project Description

~~add~~ - Review (List of Risks)

Functional Nonfunctional

+

+

V

12

74

Req. 001	Fin. ex. npx req.	L3	Fin. npx

Cancel Export

Functionen

Nonfunctional

Req. 001	- "Fill example req."	1.3	Finish
----------	-----------------------	-----	--------

Cancel

EXPORT

10/01
1.1.1

$$\begin{array}{r} 1 \\ 2 \\ 3 \end{array} \quad \begin{array}{r} 1.1 \\ 2.1 \\ 3.1 \end{array} \quad \begin{array}{r} 1.1.1 \\ 2.1.1 \\ 3.1.1 \end{array}$$

20 20.1 20.1.1

Risks Teams NonFunc Func

Project

PK	Name	Desc	FK

Risks

PK	FK	Name	Status	Priority

employee

PK	FK	Name	Sub Title

Requirements				
PK	FK	Name	Desc.	Status / Priority

- 1) Log-in
- 2) Welcome / Home
- 3) View Project(s)
- 4) New Projects
- 5?) View Requirements

[illegible]

-
- Hand-drawn sketch of a software interface for a project management tool. The interface features a header bar with a hamburger menu icon, the text "SPM", and window control icons (minimize, maximize, close). Below the header, on the left, is a "Continue:" label above a rounded rectangle containing "Title" and "description", with "DNDMA" written next to it. Below this is a "New Project" button with a plus icon. On the right, under the heading "Projects", is a table with columns "Title" and "description". The table has 8 rows, with the first row containing "DNDMA" and "description". The table is styled with a vertical scrollbar on the right and a horizontal scrollbar at the bottom.



- 1) Log-in
- 2) Welcome/Home
- 3) View Project(s)
- 4) New Projects
- 5?) View Requirements

A hand-drawn sketch of a web application interface. At the top, there's a header bar with a hamburger menu icon on the left, the text "SPM" in the center, and a user profile icon on the right. Below the header, on the left, is a login section with a circular icon containing a person silhouette, followed by the text "Welcome, <User>!". Underneath this is a "Continue:" label above a rounded rectangular button labeled "DM/M". To the right of the login section is a "Projects" section. It features a table with two columns: "Title" and "DD/MM". The table contains seven rows of wavy lines representing project entries. Below the table is a "New Project" button with a plus icon.

Create Table User

```
Primary Key UserID,  
Username Varchar(5),  
Password Varchar(8),  
);
```

Create Table Employee

```
Primary Key EmployeeID,  
Employment Varchar(50),  
JobTitle Varchar(50),  
Foreign Key (ProjectID) references Project(ID);  
);
```

Create Table Project

```
Primary Key ProjectID,  
Name Varchar(50),  
Descr Varchar(255),  
Foreign Key UserID references User(UserID),  
);
```

Create Table Risk

```
Primary Key RiskID,  
Name Varchar(50),  
Descr Varchar(255),  
Foreign Key ProjectID references  
Project(ProjectID),  
);
```

Create Table Requirement

```
Primary Key ReqID,  
Name Varchar(50),  
Descr Varchar(255),  
Status bit (bool),  
Priority int,  
Foreign Key ProjectID references Project(ProjectID),  
);
```

Create Table NERequirement

```
Primary Key NEReqID,
```

