

Kingdom of Saudi Arabia
Ministry of Education
Imam Abdulrahman bin Faisal University
College of Science and Humanities-Jubail
Computer science Department



Software Engineering
Term 1 – 2025/2026

Software Project Management Plans

For

فَضْلُ ظَهْرٍ

Version 1

CS 411 – Software Engineering
Group 2

Subervised by:

Dr. Enas El-sayed El-sharawy

2 October 2025

This Software Project Management Plans (SPMP) was prepared and provided as a deliverable for [Software Engineering, CS 411, 2025-2026], and it will be used by [CS students].

This document is based in part on the IEEE Recommended Practice for SPMP Descriptions.

#	Name	ID	Role
1	Jood Mutlaq Abaqmi	2230000478	Leader
2	Nada Ali Alshamsi	2230001749	Member
3	Jana Fahad Alharbi	2230004859	Member
4	Zahraa Mohammed Alqattan	2230006222	Member
5	Batool Sadiq Alsharqi	2230000960	Member
6	Kadijah Yasser Al bader	2230006906	Member
7	Roaa abbas Alhaddad	2230000387	Member
8	Shaden Alghamdi	2230003568	Member

Table of Content

Revision History	4
Table of Tables	5
Table of Figures	6
1. Project Overview	7
1.1 Purpose, Scope, and Objectives	7
1.2 Assumptions, Constraints and Risks	7
1.3 Project Deliverables	8
1.4 Schedule and Budget Summary	8
1.5 Evolution of the Plan	8
1.6 References	10
1.7 Definitions and Acronyms	12
1.8 Document Structure	13
2. Project Organization	14
2.1 External Interfaces	14
2.2 Internal Structure	16
2.3 Roles and Responsibilities	17
3. Managerial Process Plans	19
3.1 Start-up Plan	19
3.1.1 Estimates	19
3.1.2 Staffing	20
3.1.3 Project Staff Training	21
3.2 Work Plan	22
3.2.1 Work Breakdown Structure	22
3.2.2 Schedule Allocation	22
3.2.3 Resource Allocation	23
3.2.4 Budget Allocation	23
3.3 Project Tracking Plan	24
3.3.1 Requirements Management	24
3.3.2 Schedule Control	24
3.3.4 Quality Control	25
3.3.5 Reporting	25
3.3.6 Project Metrics	26
3.4 Risk Management Plan	27
3.5 Project Closeout Plan	27
4. Technical Process Plans	28

4.1 Process Model	28
4.2 Methods, Tools, and Techniques.....	28
4.3 Infrastructure	29
4.4 Product Acceptance	30
5. Supporting Process Plans	32
5.1 Documentation.....	32
6. Additional Plans	33

Revision History

Name	Date	Reason For Changes	Version
All members	Sep 30, 2025	initial version	0.1
All members	Oct. 7, 2025	Added project overview	0.2
All members	Oct 10, 2025	Added project organization	0.3
All members	Oct 12, 2025	Added managerial process plan	0.4
All members	Oct 13, 2025	Update 3.4	0.5
All members	Oct 14, 2025	Update 4.4	0.6
All members	Oct 15, 2025	Update 6	0.7
All members	Oct 16, 2025	Update 1.6	0.8
All members	Oct 17, 2025	Complete review	0.9
All members	Oct 18, 2025	Final version	1.0

Table of Tables

Table 1-Definitions And Acronyms.....	11
Table 2- Roles and Responsibilities.....	17
Table 3- Schedule Estimates.....	18
Table 4- Team Composition and Skills	19
Table 5- Types of Training	20
Table 6- Schedule Allocation	21
Table 7- Team member	22
Table 8- Risk Management.....	26
Table 9- project infrastructure.....	29
Table 10- project Documentation plan.....	32

Table of Figures

Figure 1- Document Structure.....	12
Figure 2- organizational chart	14
Figure 3- Type of Acceptance testing	31

1. Project Overview

1.1 Purpose, Scope, and Objectives

- This project aims to develop a safe and user-friendly mobile application allowing users to rent or share cars securely. The system aims to utilize existing vehicles more efficiently by promoting sustainable resource sharing within the local community.
- The project will not include large-scale commercial fleet operations, advanced analytics, or real-time roadside assistance.
- The application is limited to academic implementation using simulated APIs for Absher verification and payment/insurance services.
- Goals: to create a car-sharing application with core functionality, user verification, and booking features, allowing users to rate each other and contribute to environmental sustainability.
- Deliverables: prototype mobile app, documentation, and final presentation.
- Assessment: success will be determined by proper functionality, secure performance, and achievement of at least 80% successful task completion in trial runs.
- This project aligns with national digital mobility efforts and may integrate with Saudi platforms such as Absher to enhance user verification and trust.
- All requirements are defined in the approved Project Proposal Form “فَضْلُ ظَهْرٍ”, Imam Abdulrahman bin Faisal University, Term 1 – 2024/2025.

1.2 Assumptions, Constraints and Risks

- The project assumes that users possess smartphones with internet access, Absher APIs are available, and participants act responsibly.
- Constraints include limited time (one semester), limited resources, simulated integrations, and no financial budget.
- Threats include possible schedule delays, integration issues, and data security risks during authentication and processing.

1.3 Project Deliverables

- Functional mobile application (listing, booking, payment, verification).
- Rating and feedback module for owners and renters.
- Admin management interface.
- Documentation of design, implementation, and test cases.
- Final presentation and demonstration.
- Indicate the delivery means and any special handling.
- Deliverables will be submitted electronically through the Blackboard platform in PDF, PPTX formats.

1.4 Schedule and Budget Summary

- Requirement Analysis (1 week)
- System Design (2 weeks)
- Implementation (4 weeks)
- Testing (2 weeks)
- Deployment and Evaluation (1 week)

The development will progress according to the standard software development lifecycle.

There is no financial budget since this is an academic project using open-source tools.

1.5 Evolution of the Plan

Plan Compliance

This document has been created in alignment with the IEEE standard for Software Project Management Plans (IEEE Std 1058-1998). All sections, headings, and subtopics correspond to the structure required by the standard, ensuring that the project is documented, tracked, and updated in a consistent and systematic manner.

Scheduled and Unscheduled Updates

Updates to this plan will occur throughout the project's life cycle to reflect progress, risk changes, or supervisor feedback.

- **Scheduled updates** will be produced at the end of each main project phase:
 1. **Requirements & Design Phase:** after finalizing the system design and UI mockups on Figma.
 2. **Implementation Phase:** after developing and integrating the main modules (e.g., authentication, booking, Absher API).
 3. **Testing & Evaluation Phase:** after completing functional and user testing.
- **Unscheduled updates** will occur whenever:
 - A significant project scope change is approved.
 - A risk event occurs that impacts schedule or resources.
 - The supervisor requests correction or modification.
 - A new technology or tool (e.g., Firebase upgrade) requires process adjustment.

Each update will include a version number, date, editor, and a short summary of changes, recorded in the document's Revision History Table.

Dissemination of Updates

All plan updates will be distributed electronically to every team member and the project supervisor through:

- The team's Google Drive shared folder labeled "*Faql Zahr – Documents*".
- The project's GitHub repository, which will hold tagged versions of this plan (v1.0, v1.1, etc.).
- Notifications shared through the team's Trello Workspace and WhatsApp group to ensure all members acknowledge the latest version.

During the weekly meeting with the supervisor, any newly issued version of the plan will be reviewed to confirm understanding and agreement across the team.

Configuration Management of the Initial Version

- The initial version of this document (**Version 1.0**) will be uploaded to the shared Google Drive and GitHub repository once approved by the supervisor.
- This version becomes the **baseline plan**, which will remain under configuration management control.
- Only the team leader will have editing privileges, while all other members will have read-only access to prevent accidental modification.
- Any further revision must be approved before merging into the baseline folder.

Change Control After Issue

All changes to this plan after baseline approval will follow a formal change-control process, which includes:

1. **Change Proposal:** A team member identifies the required modification and fills out a *Change Request Form* (CRF) specifying the reason and impact.
2. **Review and Approval:** The *team leader* reviews the change and submits it to the *supervisor* for approval.
3. **Implementation:** Upon approval, the leader updates the document, increments the version number, and records the change in the Revision History Table.
4. **Distribution:** The updated version is re-uploaded to Google Drive and GitHub, and a notification is sent to all members.

1.6 References

2.1 Standards and Guidelines

IEEE Std 1058-1998 – *IEEE Standard for Software Project Management Plans*.

The Institute of Electrical and Electronics Engineers, Inc., Approved 8 December 1998.

ISBN 0-7381-1447-2 (Print), 0-7381-1448-0 (PDF).

Publisher: IEEE Computer Society, New York, USA.

IEEE Std 830-1998 – *IEEE Recommended Practice for Software Requirements Specifications*.

The Institute of Electrical and Electronics Engineers, Inc., 1998.

Publisher: IEEE Computer Society, New York, USA.

2.2 University Document

Project Proposal Form “فضلٌ ظهر” – *Imam Abdulrahman bin Faisal University*, Term 1 – 2024/2025.

Prepared by: Department of Computer Science and Information Technology.

Issued: September 2024.

2.3 Methodology Reference

- **Agile Methodology Guide – Scrum Framework Overview.**

Version 3.0, 2024.

Prepared by: Software Engineering Course Materials, IAU.

Publisher: Department of Software Engineering.

2.4 Online and Research Sources

- **GeeksforGeeks.** *Acceptance Testing – Software Testing.*
Available at: <https://www.geeksforgeeks.org/software-engineering/acceptance-testing-software-testing/>
Date Accessed: October 2025.
- **ArXiv.org.** *Software Engineering: A Review of Agile and Traditional Methodologies.*
ArXiv Identifier: 2101.02361, January 2021.
Authors: Various. Publisher: Cornell University Library.
- **ScienceDirect.** *An Empirical Study of Software Project Management Practices. Information and Software Technology*, Elsevier, Volume 56, Issue 5, 2014.
Accessed: October 2025.
- **IEEE Xplore.** *IEEE Standard for Software Project Management Plans.*
Document Number: 1715481.
DOI: 10.1109/IEEESTD.1998.1715481.
Accessed: October 2025.
- **Wiley Online Library.** *Software Maintenance and Evolution: Research and Practice.*
DOI: 10.1002/smr.2297. Published 2021.
Accessed: October 2025.

2.5 Deviations from Referenced Standards

- This Software Project Management Plan conforms to the structure and content requirements of **IEEE Std 1058-1998**. However, the following justified deviations have been applied:
- **Adoption of Agile Scrum framework:** Iterative planning and review cycles are used instead of a strictly sequential (Waterfall) process.
- **Electronic documentation:** All project documents are maintained digitally within version-controlled repositories instead of printed baselines.
- **Simplified structure:** Some IEEE sub-clauses are merged to fit the academic SPMP template provided by the university.

1.7 Definitions and Acronyms

Term / Acronym	Definition / Description
IEEE	Institute of Electrical and Electronics Engineers – provides the standard (IEEE 1058) followed for this plan.
SPMP	Software Project Management Plan – the main document outlining project organization and management.
UI/UX	User Interface / User Experience – focuses on design and usability aspects of the mobile application.
API	Application Programming Interface – allows connection between the app and external systems (e.g., Absher).
Absher	Saudi government platform used for digital identity verification.
Firebase	Google cloud service used for authentication, data storage, and hosting.
Flutter	Google’s open-source framework used to develop the Faḍl Zahr mobile app.
GitHub	Version control platform used by the team to store and manage code and documents.
Trello	Online project management tool used for tracking tasks and team progress.
Gantt Chart	A visual tool representing the project schedule, milestones, and task duration.
QA	Quality Assurance – ensures all deliverables meet project quality standards before submission.
Version Control	The process of managing changes to documents or code using tools like GitHub.
Baseline	The approved version of this plan (Version 1.0) that can only be modified through formal change control.

Table 1- Definitions And Acronyms

1.8 Document Structure

Table of Content

Revision History	4
Table of Tables.....	5
Table of Figures	6
1. Project Overview	7
1.1 Purpose, Scope, and Objectives	7
1.2 Assumptions, Constraints and Risks	7
1.3 Project Deliverables.....	8
1.4 Schedule and Budget Summary	8
1.5 Evolution of the Plan	8
1.6 References	10
1.7 Definitions and Acronyms.....	12
1.8 Document Structure	13
2. Project Organization	14
2.1 External Interfaces	14
2.2 Internal Structure.....	16
2.3 Roles and Responsibilities.....	17
3. Managerial Process Plans	19
3.1 Start-up Plan	19
3.1.1 Estimates.....	19
3.1.2 Staffing	20
3.1.3 Project Staff Training	21
3.2 Work Plan	22
3.2.1 Work Breakdown Structure	22
3.2.2 Schedule Allocation	22
3.2.3 Resource Allocation	23
3.2.4 Budget Allocation	23
3.3 Project Tracking Plan.....	24
3.3.1 Requirements Management	24
3.3.2 Schedule Control	24
3.3.4 Quality Control	25
3.3.5 Reporting	25
3.3.6 Project Metrics.....	26
3.4 Risk Management Plan.....	27
3.5 Project Closeout Plan	27
4. Technical Process Plans	28
4.1 Process Model.....	28
4.2 Methods, Tools, and Techniques.....	28
4.3 Infrastructure	29
4.4 Product Acceptance.....	30
5. Supporting Process Plans	32
5.1 Documentation	32
6. Additional Plans.....	33

Figure 1 - Document Structure

2. Project Organization

2.1 External Interfaces

1. Describe the organizational boundaries between the project and external entities:

It has been explained that the project interacts with several external entities, including:

- **The Parent Organization:** The organization that manages and supports the project financially and technically.
- **The Customer:** The individuals who need a car for short-term rental.
- **Subcontracted Organizations:** Other companies that provide additional services, such as insurance, payment processing, or data storage.
- **Other Organizational Entities:** Entities such as **Absher** for identity verification or logistics companies that manage shared private vehicles.

3. Identify, as applicable:

- **The Parent Organization:** It was clarified that the parent organization could be a software development firm or a tech company providing financial and technical support.
- **The Customer:** The customer is identified as the individual who needs a car for temporary rental.
- **Subcontracted Organizations:** It was mentioned that the project might partner with other organizations to provide services like insurance, payment processing, or data hosting.
- **Other Organizational Entities:** Entities like **Absher** for identity verification and logistics companies handling private cars available for sharing.

3. Use organizational charts or diagrams to depict the project's external interfaces:

It has been explained how to represent the relationships between the project and external entities using an organizational chart:

- **The customer** interacts with the **app** for booking and payment.
- **Vendors** provide services like insurance and payment processing.
- **Government Entities** such as **Absher** contribute to the security verification process.



Figure 2 – organizational chart

- **Parent Org:** The main organization responsible for overseeing the project, such as a software development or tech company providing financial and technical support.
- **Fadl Dhar Project:** The project itself, which facilitates car-sharing between individuals. This is the central part of the diagram.
- **Customers:** Individuals who use the app to book cars and make payments.
- **Vendors:** External companies that provide essential services like **insurance**, **payment processing**, and other logistical services.
- **Government Entities (Absher):** Government organizations like **Absher** contribute to the security verification process by confirming user identity, ensuring that only authorized users are involved in the transaction.
- **Other Entities:** This could be any other external organization involved in supporting the project, such as logistics companies managing vehicle availability.

2.2 Internal Structure

- **Internal organization of the project - Team structure and different units of development:**

Design Team: Responsible for creating the app and user interfaces.

Development Team: Responsible for writing the code and configuring the environment for the application.

Testing Team: Ensures that the app operates without any glitches and is error-free.

Technical Support and Maintenance Team: Handles the ongoing support and maintenance of the app after deployment.

- **Interfaces Between Development Units:**

- It is essential for the Design Team and Development Team to maintain close communication to ensure the final app satisfies all its features.
- It is necessary for the Development Team and the Testing Team to work together to ensure that the code is free of bugs, which is crucial for the app's functionality.

- **Supporting Processes:**

Configuration Management: Managed by the Development Team to maintain the app's functionality and ensure its features are kept up to date.

Quality Assurance (QA): Managed by the QA Team to maintain the app's quality by checking for defects and ensuring it meets the required standards.

Verification and Validation: The Validation Team, typically overseen by admins, performs these tasks to ensure the app meets customer requirements and functions as expected.

- **Internal Structure Diagram:**

- An internal organizational chart would be created to document the development team's activities. This chart would break down the units in charge of configuration management, QA (maintenance), and verification, along with associated processes. The diagram would also display the interfaces and communication channels between the units, with lines indicating the relationships and interactions between teams (Design, Development, Testing, and Validation).

2.3 Roles and Responsibilities

Major Work Activities:

Software Design: The design team specifies the system architecture, user interface, and software structure. This is their primary responsibility.

Design Specifications: These specifications are used to implement the code, which is overseen by the development team. The job can be classified as coding/development as well.

Testing: The process of ensuring that the software meets functional requirements and eliminates errors.

Deployment: The deployment team is responsible for transporting app components to customers' locations.

Management: Managed by the documentation team, which compiles user manuals, code documentation, and system specifications.

Supporting Processes:

Configuration Management: Managed by CMIL. The management of configuration.

Quality Assurance: Managed by the Quality Assurance team.

Verification and Validation: Managed by the verification and validation team.

Organizational Units Responsible for Activities:

Planning: Managed by the design team.

Development: Overseen by the development group.

Execution: Managed by the testing teams.

Support: Managed by the support team.

Work Activity	Design team	Development team	Testing team	Deployment team	Support team	QA team	Validation Team
Software Design	Responsible		--	--	--	--	--
Coding/Development	--	Responsible	--	--	--	--	--
Testing	--	--	Responsible	--	--	--	--
Deployment	--	--	--	Responsible	--	--	--
Documentation	--	--	--	--	Responsible	--	--
Configuration Management	--	--	--	Responsible	--	--	--
Quality Assurance (QA)	--	--	--	--	--	Responsible	--
Verification & Validation	--	--	--	--	--	--	Responsible

Table 2 - Roles and Responsibilities

Work Activities: The major activities in the project include Software Design, Coding/Development, Testing, Deployment, and more.

Design Team: Responsible for Software Design, including creating system architecture and user interfaces.

Development Team: Responsible for writing code and implementing the software according to the design specifications.

Testing Team: Responsible for Testing the software to ensure that it meets the functional requirements and is free from errors.

Deployment Team: Responsible for Deploying the app to the customer's environment or production.

Support Team: Handles Support and Maintenance for the app after deployment.

QA Team: Responsible for Quality Assurance (QA), ensuring that the app meets the required quality standards.

Validation Team: Responsible for Verification and Validation to ensure that the app meets customer requirements and works as expected.

3. Managerial Process Plans

3.1 Start-up Plan

3.1.1 Estimates

- **Cost Estimates:**

- There are no direct financial costs since the project is an academic one.
- All tools and resources are free or open source.

- The main technical resources include:

Development Environment: Android Studio
 Version Control: GitHub
 Database: Firebase (free tier) or local database
 Design Tools: Figma, Canva
 Project Management Tools: Trello or Notion

- **Schedule Estimates:**

The project is scheduled to be completed within the academic semester (15 weeks) The detailed schedule is as follows:

Phase	Duration	Description
Requirements Analysis	Weeks 1–2	Identify and document functional and non-functional requirements.
Design	Weeks 3–5	Create the overall system architecture and user interface design.
Implementation	Weeks 6–10	Develop and integrate all system modules.
Testing	Weeks 11–13	Perform functionality, performance, and security testing.
Deployment & Final Presentation	Weeks 14–15	Finalize, prepare documentation, and present the project.

Table 3 – Schedule Estimates

- **Resource Estimates:**

- The team consists of 8 students from Computer Science department
- All devices and technical tools are available to the team members
- No need for external financial or contractual resources

- **Confidence Levels:**

The confidence level is **medium to high (~85%)**, based on previous academic project data and regular supervisory follow-up.

- **Estimation Methods and Sources:**

- **Analogy Estimation** :Based on data and schedules from previous student projects.
- **Rule of Thumb** :Workload and duration estimated according to instructor experience and team size.

- **Re-estimation Plan:**

- Toole for Re-estimation: Gantt charts, Trello boards, and Weekly progress reports
- Schedule: Re-estimation will occur usually every week or at major milestones

3.1.2 Staffing

Role	Required Skills	Main Tasks	Project Phase	Duration
Project Leader	Leadership, time management	Coordinate work, communicate with supervisor	All phases	15 weeks
System Analyst	Requirements analysis	Gather and document system requirements	Analysis	2 weeks
UI/UX Designer	User interface and experience design	Design application screens and flow	Design	3 weeks
Front-End Developer	Android development, API integration	Build and connect the user interface	Implementation	5 weeks
Back-End Developers (2)	Database design, logic implementation	Develop and connect backend services	Implementation & Testing	8 weeks
QA & Testing Specialist	Testing, debugging, quality assurance	Plan and execute test cases	Testing	3 weeks
Documentation & Presentation Officer	Report writing, presentation skills	Prepare documentation and final presentation	Deployment & Closure	2 weeks

Table 4 - Team Composition and Skills

- **Source of Personnel**

All team members are students in Computer Sciences program
No need for recruitment or contacting

- **Duration of Resource Usage**

Participation ranges from **phase-based (2–5 weeks)** for roles like analysis or testing, to **full-term (15 weeks)** for administrative and leadership roles.

- **Resource Representation**

We will use Garntt chart to visualize the staffing schedule by skill level and project phase

3.1.3 Project Staff Training

Training Type	Skills to be Developed	Training Method	No. of Trainees
Technical	Android development, GitHub, APIs	Self-paced learning + practice	8
Managerial	Time management, task tracking, documentation	Instructor supervision + Trello	8
Supporting	Report writing, presentation skills	Group sessions + feedback from supervisor	8

Table 5 - Types of Training

- **Training Methods and Tools**

- Approaches: online tutorials, academic mentoring, hand-on practice
- Resources: YouTube, Coursera

- **Training Scope**

- **Technical Skill:** programming, database and app development
- **Managerial Skills:** planning, task coordination and progress tracking
- **Supporting Skills:** presentation, communication, documentation

3.2 Work Plan

3.2.1 Work Breakdown Structure

The project “**Faḍl Zahr**” will be divided into several main phases as follows:

1. **Requirements Analysis:**

- Gather requirements from both car owners and renters through interviews and surveys.
- Define system features such as vehicle listing, booking, payment, verification, and rating.

2. **System Design:**

- Design the database schema, system architecture, and UI mockups.
- Specify workflows for car listing, booking, and user verification.

3. **Implementation:**

- Develop the mobile application using appropriate technologies.
- Implement user authentication, Absher verification, and payment gateway integration.

4. **Testing:**

- Perform unit, integration, and user acceptance testing.
- Verify that all main features (listing, booking, payment, rating) function correctly.

5. **Deployment and Documentation:**

- Deploy the prototype version for demonstration.
- Prepare technical documentation, user manual, and testing reports.

3.2.2 Schedule Allocation

The project is planned to be completed over **15 weeks**, following this timeline:

Phase	Duration	Start	End
Requirements Analysis	2 weeks	Week 1	Week 2
System Design	3 weeks	Week 3	Week 5
Implementation	5 weeks	Week 6	Week 10
Testing	3 week	Week 11	Week 13
Deployment & Final Report	2 week	Week 14	Week 15

Table 6 - Schedule Allocation

3.2.3 Resource Allocation

- **Team Members:**

<i>Jood Mutlaq Albaqmi</i>	Project Leader (planning and coordination)
<i>Nada Ali Alshamsi</i>	Front-end development
<i>Roaa Abbas Alhaddad</i>	Back-end and database design
<i>Jana Fahad Alharbi</i>	Mobile app development
<i>Zahraa Mohammed Alqattan</i>	Testing and documentation
<i>Batool Sadiq Alsharqi</i>	UI/UX design and user experience testing
<i>Kadijah Yasser Al Bader</i>	Security and Absher integration
<i>Shaden Alghamdi</i>	Payment integration and deployment support

Table 7 – Team member

- **Tools and Resources:**

- Flutter (for mobile app development)
- Firebase / MySQL (for backend and database)
- Figma (for UI design)
- GitHub (for version control)
- Trello (for task tracking and collaboration)

3.2.4 Budget Allocation

- No financial budget is required. However, the following resources will be used:
- Students' personal laptops and mobile devices
- Students' time and effort
- Free development and design tools (Flutter, Firebase, Figma, GitHub)
- Internet access for team collaboration and testing

3.3 Project Tracking Plan

3.3.1 Requirements Management

- All requirements for the “*Fadl Zahr*” mobile application, such as user login, Absher verification, car listing, reservation, payment, rating, and trip tracking, will be tracked in a shared Google Sheet. Each requirement will have an ID, short description, responsible member, and progress status to ensure all parts of the system are implemented correctly.
- Any new feature or change like updating the interface or adding a booking option will go through a simple Change Request Form that explains the reason and possible impact. The team leader and requirements manager will review it before approval, and approved changes will be added to the shared document to keep everyone informed.
- Before accepting a change, the team will analyze its effect on the project’s scope, quality, schedule, and resources. If a change could delay progress or affect user experience, the plan and timeline will be adjusted to avoid problems.
- Major updates will be discussed in a short team meeting with the supervisor before implementation. A small group consisting of the team leader, requirements manager, and database developer will act as the Change Control Board (CCB). They will review and approve requirement updates, ensuring each aligns with the project goals. The requirements manager will update the Google Sheet and save versioned copies in Google Drive for reference.
- To keep requirements organized, the team will use a Requirement Traceability Matrix (RTM) linking each requirement to its related design, code, and test. Early interface prototypes created in Figma will help visualize the design and confirm it meets user needs. Regular review meetings and impact analysis will ensure all requirements stay valid, consistent with the original plan, and meet quality standards.

3.3.2 Schedule Control

- The schedule for the “*Fadl Zahr*” project will be developed and monitored as part of the project management process. A detailed timeline will include all major phases such as analysis, design, implementation, testing, and final presentation. Each activity will have a defined start and end date, an assigned team member, and a clear expected deliverable.
- Schedule control will focus on comparing planned progress with actual progress during review meetings. At each milestone, progress will be measured based on the percentage of

completed tasks and the quality of submitted deliverables. If any delay or deviation occurs, corrective actions such as adjusting task priorities or extending deadlines will be applied to maintain the schedule.

- Tools such as Google Sheets and Gantt Charts will be used to track, measure, and update the schedule regularly. Each phase will be considered complete once its deliverables, including reports, designs, and test results, meet the required quality standards.
- This process helps ensure that the “Fadl Zahr” project remains organized, on time, and aligned with its overall objectives and timeline.

3.3.4 Quality Control

- Quality control for the “*Fadl Zahr*” project will focus on maintaining the accuracy, consistency, and reliability of all deliverables throughout the project lifecycle. Each development phase analysis, design, implementation, and testing will include review steps to verify that the outputs meet the required standards and project objectives.
- The team leader and the quality reviewer will evaluate the deliverables using predefined checklists and quality criteria. Peer reviews will be conducted after each major milestone to detect errors early and ensure that the work aligns with the system requirements. If any issue or defect is identified, corrective actions will be taken immediately, and the affected component will be rechecked before moving to the next phase.
- Testing activities will include both functional and non-functional evaluations to confirm that the system operates correctly, provides accurate results, and maintains good performance. All test results and review notes will be documented and stored for tracking and improvement purposes.
- This process helps ensure that the “*Fadl Zahr*” project maintains high quality, meets user expectations, and achieves the planned goals effectively.

3.3.5 Reporting

- Project reporting in the “*Fadl Zahr*” project will ensure that all team members and the supervisor stay informed about progress, issues, and upcoming tasks.
- Regular reports will summarize the status of activities, completed milestones, pending tasks, and any challenges that may affect the project schedule or quality.

- Weekly progress reports will be prepared by the team leader and shared with all members through the project's shared folder. These reports will highlight completed work, upcoming deadlines, and any deviations from the original plan. If any issues arise, they will be discussed in weekly meetings to identify corrective actions and update the project schedule accordingly.
- At the end of each major phase such as design or testing a summary report will be submitted to the supervisor, including achievements, problems encountered, and recommendations for improvement.
- All reports will be stored in the team's shared drive to maintain proper documentation and provide clear evidence of project tracking and management.
- This reporting process helps maintain transparency, supports decision-making, and ensures that the "*Fadl Zahr*" project stays aligned with its objectives and planned timeline.

3.3.6 Project Metrics

- Project metrics for the "*Fadl Zahr*" project will be used to measure progress, performance, and overall success throughout all phases. These metrics will help the team evaluate how well the project follows its plan in terms of schedule, quality, and task completion.
- Key metrics will include:
 - **Schedule Performance:** comparing planned versus actual completion dates for each milestone.
 - **Task Completion Rate:** measuring the percentage of finished activities relative to the total number of tasks.
 - **Quality Metrics:** tracking the number of errors, rework items, or defects found during testing and review.
 - **Team Efficiency:** evaluating how effectively team members complete their assigned tasks on time.
- The team leader will review these metrics during weekly meetings to identify any issues or delays and take corrective actions when needed.
All collected data will be recorded in a shared document to track improvements and maintain transparency in project progress.
- By using these metrics, the "*Fadl Zahr*" team can ensure that the project remains on schedule, maintains high quality, and achieves its objectives efficiently.

3.4 Risk Management Plan

The Risk Management Plan identifies potential risks that may affect the project and outlines strategies to minimize their impact

No	Risk Description	Probability	Impact	Preventive Action	Contingency Plan	Responsible Person
1	Delay in project deliverables	High	Medium	Prepare a detailed schedule and assign clear responsibilities	Extend the delivery deadline if necessary	Project Manager
2	Technical issues during development	Medium	High	Conduct regular testing and use reliable tools	Allocate extra time for debugging	Technical Team
3	Lack of team communication	Medium	Medium	Hold weekly meetings and use online collaboration tools	Assign a communication coordinator	Team Leader
4	Insufficient resources or equipment	Low	High	Ensure early resource planning	Rent or borrow required equipment	Project Manager

Table 8 - Risk Management

The project team will review risks weekly and update the plan as the project progresses.

3.5 Project Closeout Plan

The Project Closeout Plan defines the steps to formally complete and close the project:

1. **Deliverables Handover:** Final project outputs will be delivered to the client or supervisor.
2. **Final Evaluation:** The project performance will be evaluated against objectives and requirements.
3. **Documentation:** All reports, designs, and technical files will be archived.
4. **Lessons Learned:** The team will hold a meeting to discuss lessons learned and best practices.
5. **Resource Release:** Team members and resources will be released from project duties.
6. **Project Closure Report:** A final closure document will be submitted to confirm the project's completion.

4. Technical Process Plans

4.1 Process Model

- This project follows the Agile Process Model, which allows flexibility and continuous improvement throughout development.
- The work is divided into short iterations called sprints, where each sprint includes planning, development, testing, and review phases.

Phases of the Agile Model:

1. Requirement Analysis
 2. Design and Planning
 3. Implementation (Development)
 4. Testing and Evaluation
 5. Deployment and Feedback
- This approach ensures continuous communication with stakeholders and faster adaptation to changes.

4.2 Methods, Tools, and Techniques

- Requirement analysis using interviews and brainstorming.
- System design using flowcharts and UML diagrams.
- Incremental testing after each sprint.

Tools:

- Project Management: Trello / Jira
- Design: Figma / Draw.io
- Development: Visual Studio Code / GitHub
- Documentation: Microsoft Word / Google Docs

Techniques:

- Version control using Git.
- Code review and peer evaluation.
- User testing for quality assurance.

4.3 Infrastructure

component	Specification and recommendation
Hardware	<ul style="list-style-type: none"> - Mobile devices, and laptops - Processor: Intel Core i5 - Memory (RAM): 8 GB – 16 GB - Storage: Minimum 500 GB SSD - Accessories: The built-in camera and microphone enable remote collaboration and meetings
Operating system	<ul style="list-style-type: none"> - Windows 11 and macOS - Updates: Turn on automatic updates to have all of the latest security patches
Software	<ul style="list-style-type: none"> - Office Tools: Microsoft Office Suite - Design & Collaboration Tools: Adobe Creative Cloud, Canvas - Project Management Tools: Trello or Notion for collaborative work and task management.
Network	<ul style="list-style-type: none"> - Connections: Wireless routers (Wi-Fi 4G) or (Wi-Fi 5G) support for secure and fast connections to all of our devices.
Policies & procedures	<ul style="list-style-type: none"> - Backup data: Store documents in cloud storage services, to ensure automatic backups of both documents and projects (OneDrive, Google Drive) - Password: Use strong, unique passwords for each service and store them in a password manager
Facilities	<ul style="list-style-type: none"> - Workspace: A quiet space reserved for virtual meetings and solo work. - Conditioning: Adequate lighting and air handling for comfort and productivity.

Table 9 – project infrastructure

4.4 Product Acceptance

Acceptance plan:

Customer acceptance will take place following the successful conclusion of the final system testing phase and an official showcase of the application's key functionalities, including:

- User sign-up and login (for car owners and renters).
- Management of car listings and availability.
- Processes for booking, payment, and confirmation.
- Rating and review mechanism.
- Functions for customer assistance and managing disputes.

Upon completion of these steps, a formal acceptance meeting will be conducted between the project team and the customer.

The customer will review the deliverables, test the system functionalities, and provide written confirmation of acceptance.

Acceptance Requirements:

1. All functional requirements outlined in the Software Requirements Specification (SRS) are executed and functioning properly
2. The system successfully meets all acceptance test cases without any critical or major issues
3. Performance and usability meet the agreed benchmarks.
4. The application operates reliably on supported mobile and web platforms.
5. All documentation (user manual, installation guide, and technical documentation) is complete and approved

Formal Agreement:

A formal Product Acceptance Agreement will be signed by representatives of both the IM/IT organization and the customer after successful completion of all acceptance tests. This document will confirm that all deliverables meet the agreed-upon standards and are ready for production deployment.

Acceptance Methods and Tools:

The following methods and tools will be used to verify deliverable acceptance:

- **Testing:** Run acceptance test scenarios on the completed system version
- **Display:** demonstration of important application processes.
- **Evaluation:** Analyzing test results, performance indicators, and user input.
- Examine papers to ensure conformance to design specifications.

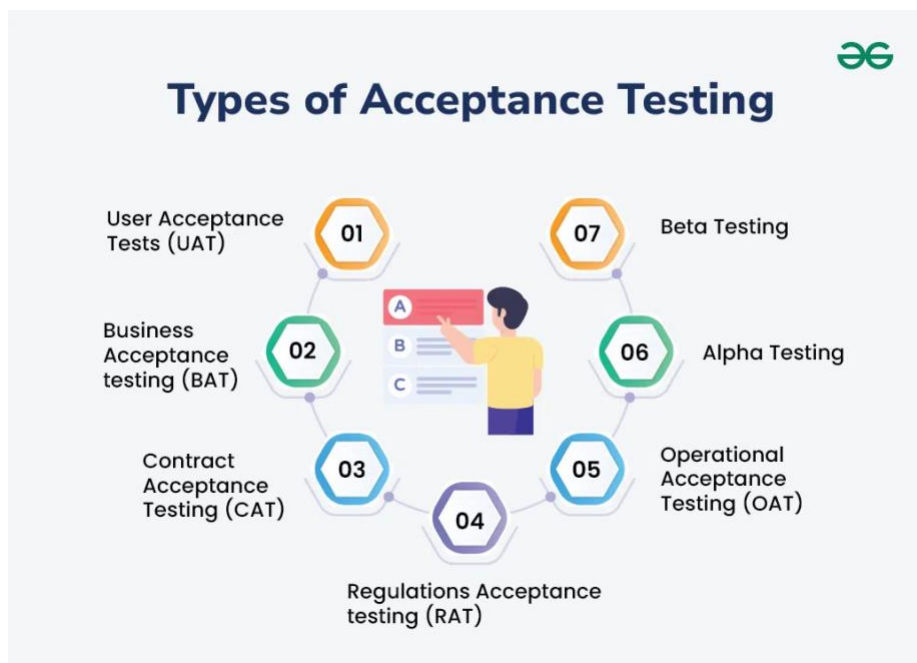


Figure 3 – Type of Acceptance testing

5. Supporting Process Plans

5.1 Documentation

Document	Template	Person Responsible for preparation	Person Responsible for Review
Project Management Plan (SPMP) Document	Standard SPMP Template	All Team Members	Academic Supervisor
Software Requirements Document (SRS)	IEEE Std 830 standard (as a theoretical reference)	Analysis and Documentation Team	Academic Supervisor, Project Manager (Team Leader)
Software Design Document (SDD)	IEEE Std 1016 Standard (For Theoretical Reference)	Design Team	Development Team, Project Manager
Software Testing Documentation (STD)	Standard IEEE Std 829	QA Team	Project Manager, Development Team
User manual	Standardized template (e.g., standardized structure, clear language)	Documentation team	Development team, representative of end users (another person from the team)
Final Status Report/Presentation	Presentation Template Specified by College	All Members by Tasks	Academic Supervisor, Project Manager

Table 10 – project Documentation plan

6. Additional Plans

In ensuring the product fulfills all contract specifications and conditions, a number of supplemental plans will be developed and adhered to in the execution of the project.

6.1 Safety, Confidentiality, and Protection:

To protect user data, system integrity, and overall security, we will continue to adhere to open protocols. This involves routine security audit procedures, using encryption protocols, and restricting access to confidential information to authorized staff.

6.2 Special Amenities or Tools:

All test, validation, or operational product specialty tools, hardware, or software will be determined and ready in advance for free flow and reproducible uniformity.

6.3 Installation of the Product:

There will be an installation step-by-step guide that will simplify installation by including system requirements, configuration information, and instructions to test that the product has been properly installed and is functional.

6.4 User Training:

Training programs will be developed to bring about significant use of the product by the users. These may be in the form of workshops, web-based tutorials, and user guides to acquaint all stakeholders with system usage.

6.5 Interfacing with Other Systems:

If the product must integrate with current systems, an integration plan will be outlined. It will include data and process integration information to ensure compatibility and simplicity of data transfer between systems.

6.6 Data Migration:

In the data conversion from legacy systems, a data conversion plan will make the process precise, secure, and keep all the data intact while switching from one system to another.

6.7 Transition System:

Transition plan will ensure the transitioning away from the old system and onto the new one with least disruption. The plan will contain backup, rollout, and testing processes to allow operations to be continued running efficiently.

6.8 Maintenance and Support:

After product acceptance has taken place, support will be continuous. It is updates, debugging, and ongoing maintenance to allow the product to remain secure, stable, and current