CST 2550 Software Engineering Management and Development

Group Coursework : Project Management

Autumn/Winter term 2024/2025

STUDENT NAMES	STUDENT ID NUMBER
AMEERA NAJLA MANJOO	M01014463
PAULINE JONAS MTALLO	M00937209
SHEJA DAVID	M00977005
MATHEWS MWANGI	M00836628

Lab Tutor: Aisha Idoo
Date of submission : 16/04/25

Table of Contents

Table of Figures

Project Description

The team was responsible for developing a Driving Lesson Booking System, which involved the development of a tailored data structure to meet the project's requirements. The system was designed to provide users with a seamless experience by incorporating the following features:

1. Displaying available driving lesson options
2. Searching for specific lessons or schedules
3. Booking lessons based on user preferences
4. Adding new lesson slots to the system
5. Removing unavailable or canceled lesson slots

To ensure the successful delivery of the project, the team focused on fostering strong collaboration and maintaining open communication. By working together efficiently, they were able to streamline the development process and achieve their goals effectively.

Team Role

The team is organized as follows:

Scrum Master: Ameera Manjoo
Secretary: Pauline Mtallo
Developers:
 - Mathews Mwangi
 - Nnamdi Nwukwa
Tester: Sheja David

Tools Used

The team stayed in regular communication through WhatsApp, where members shared updates on milestones and flagged any issues that arose. For more detailed discussions and strategic planning, we alternated between in-person meetings and virtual sessions using Google Meet. To facilitate collaboration, we utilised a shared folder in Google Docs and Notion, which allowed the team to access and update important project materials such as diagrams (including use case, class, and activity diagrams), the Work Breakdown Structure, and the Gantt Chart). Visual Studio 2022, SQL Lite Management and MSTest are environments that were put into use. The commits were eventually carried out on Github.

Notes of Meetings

To ensure effective collaboration, communication, and project planning, the team held weekly sprint meetings, with all notes uploaded to GitHub for easy access and reference. Additionally, the team conducted regular stand-up meetings each day to review progress, address any challenges encountered, and outline the next steps for tasks. These meetings helped ensure everyone stayed on track and aligned with the project goals.

Workload Division

Project Planning

The Scrum Master has been responsible for the general project management, coordination and ensure synergy among members. Meetings were set weekly and recorded by the Secretary.

A general plan was created on Notion and updated weekly based on the project advancements. Sub-section pages were created for different members to allow flow and promote clarity.

Resource management was another factor that had to be taken into consideration given that some of the members were Mac users while others were Windows users. Members were allocated tasks based on their strengths and weaknesses.

As for the commits on GitHub, members were assigned different classes of the project to code and implement. Different branches were initially created and then ultimately merged for the testing process.

Eventually, the objectives of the project were met step by step, based on small terms goals, milestones achieved and constant update for betterment.

Work Timeline

Given the fixed deadline and the change of date of submission 10 days prior to the submission, planning was important and had to be reviewed constantly. Efficiency, teamwork and time efficiency were the focus. A WBS and Gantt Chart were used for breakdown of tasks and visualization of deadline and progress made.
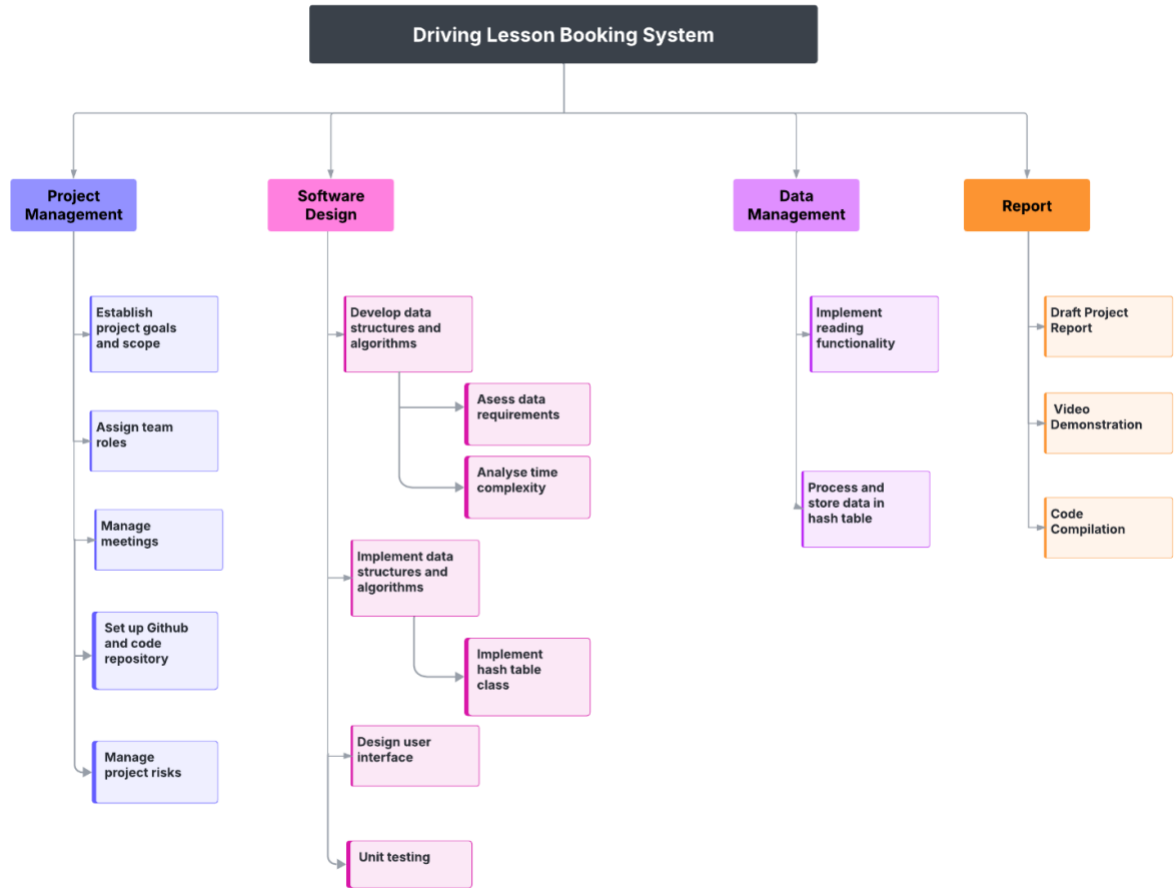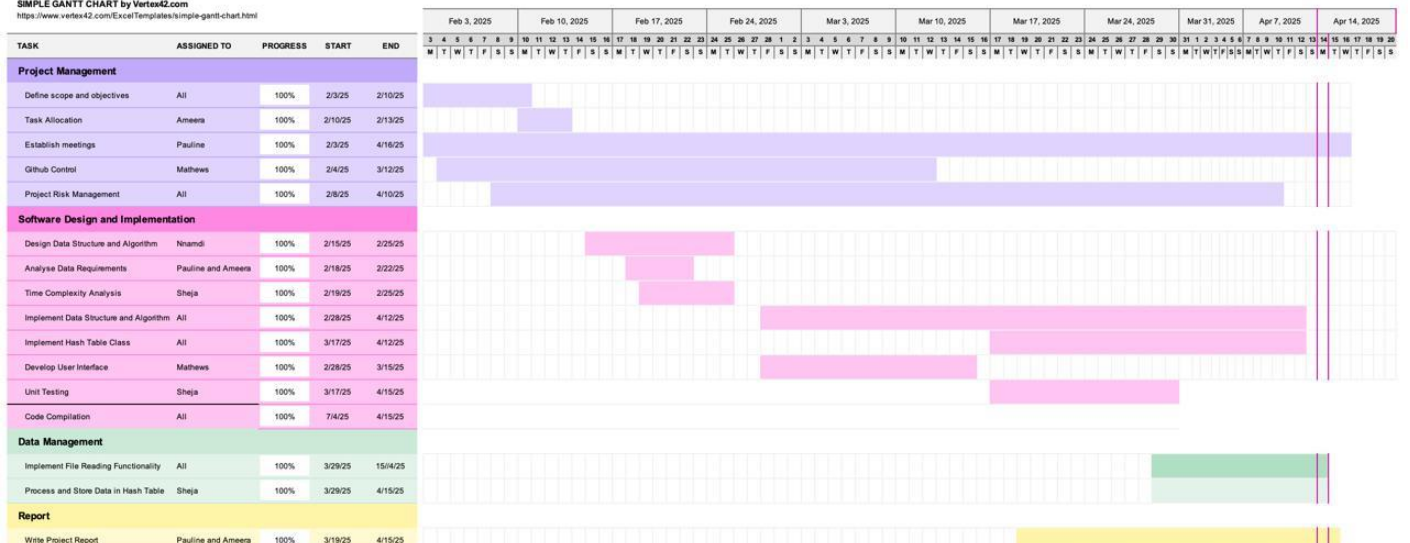
Figure 1: WBS

Figure 2: Gantt Chart

Project Design and Implementation

The design of the program was a crucial part that needed constant review and updates. The project requiring groupwork meant that it had to be equally divided between all members. To assist this process, activity diagrams were drawn only to differentiate between the different functionalities of the program but also have a diagrammatical way to ease the task division.
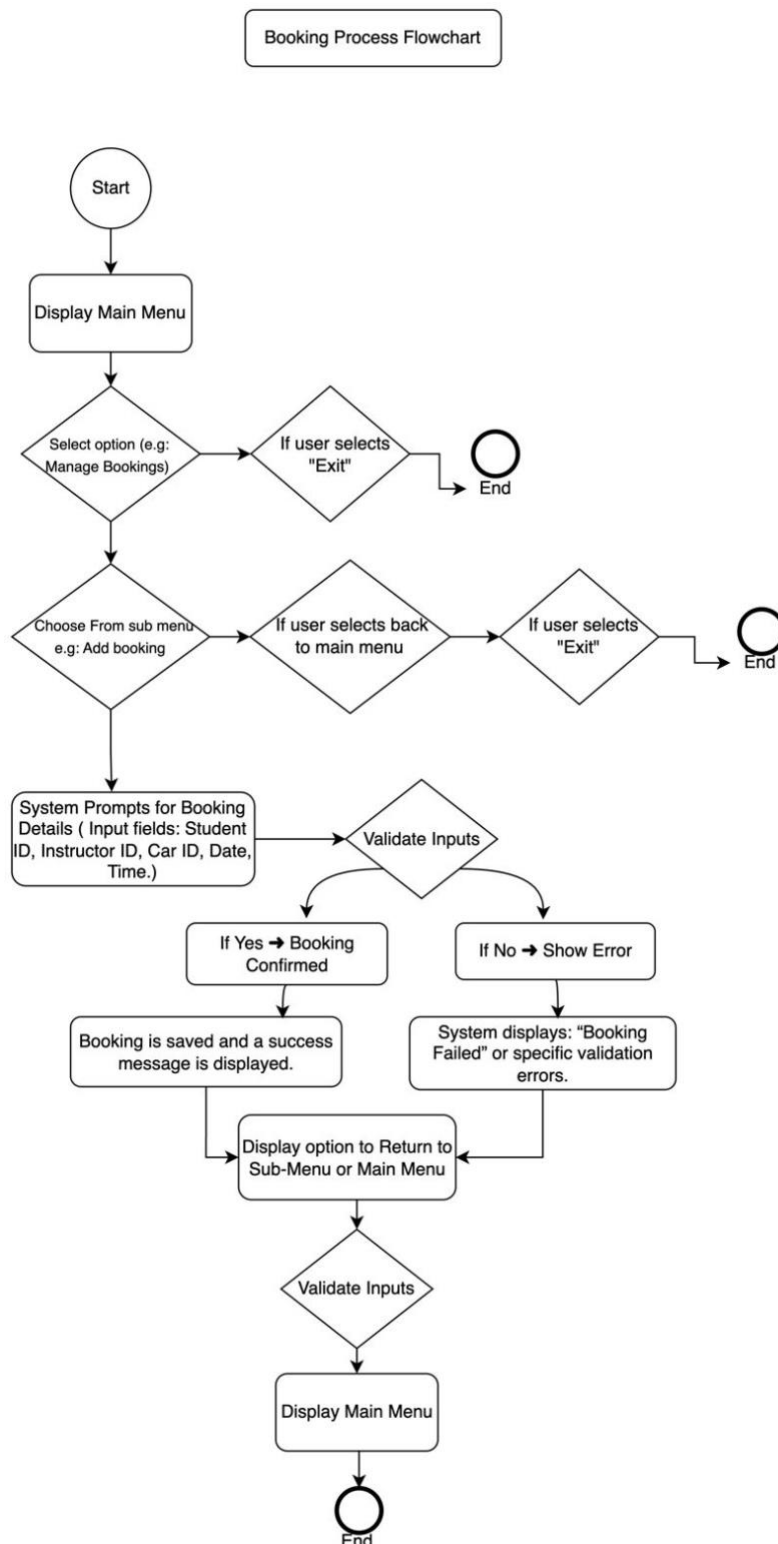

Figure 3: Booking Process Flowchart

Students Management
Flowchart

Start

Display Main Menu

Select option (e.g:
Manage Students)

Displays Manage
Bookings menu

If user selects back
to main menu

Display Main Menu

If user selects
"Exit"

End

Choose From sub menu
e.g: Delete Student

Enter Student ID to delete

Input Student ID

If ID is found Displays:
Student Deleted Succesfully

If ID is Not found Displays:
Student not found

Displays Manage
Students menu

Select option (e.g:
Display all Students)

Displays All
Students

Displays Manage
Students menu

If user selects back
to main menu

Display Main Menu

If user selects
"Exit"

End

Figure 4: Student Management Flowchart

Instructor Management Flowchart

Start

Display Main Menu

Select option (e.g: Manage Instructors)

Displays Manage Instructors menu

If user selects back to main menu

Display Main Menu

If user selects "Exit"

End

Choose From sub menu e.g: Delete Instructors

Enter Instructors ID to delete

Input Instructors ID

If ID is found Displays: Instructor Deleted Successfully

If ID is Not found Displays: Instructor not found

Displays Manage Instructors menu

Select option (e.g: Display all Instructors)

Displays All Instructors

Displays Manage Instructors menu

If user selects back to main menu

Display Main Menu

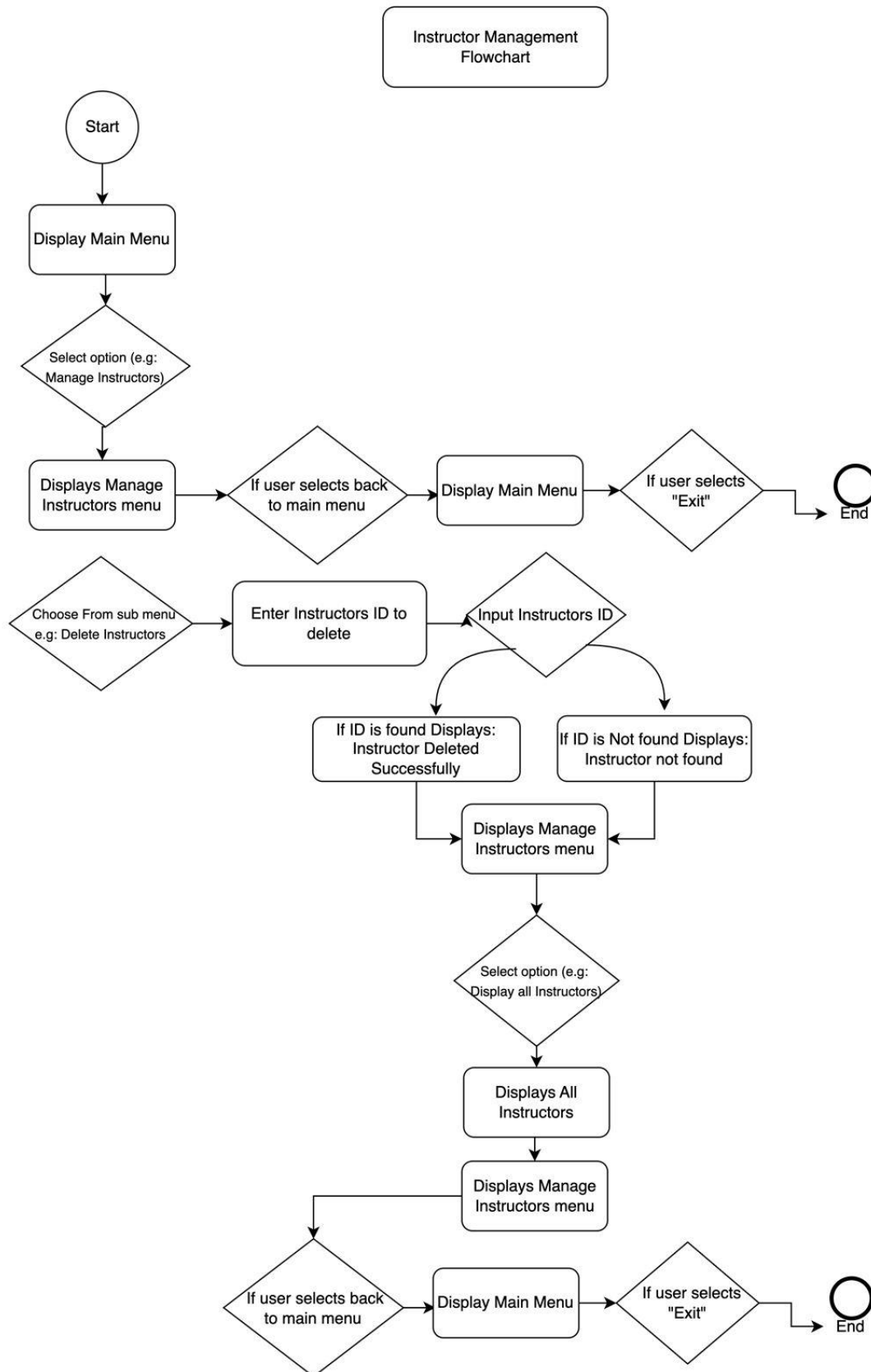If user selects "Exit"

End

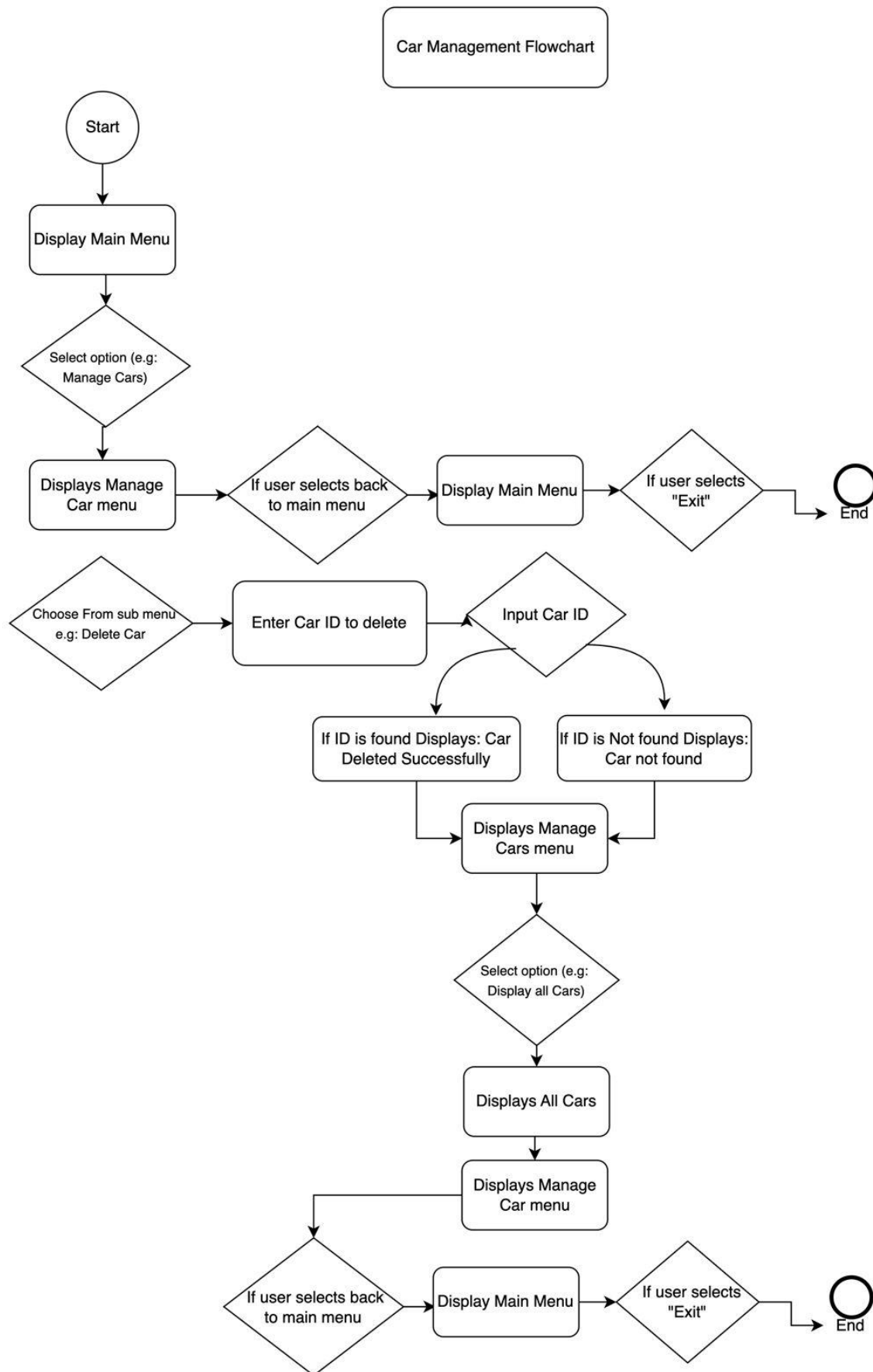*Figure 5: Instructor Management Flowchart*
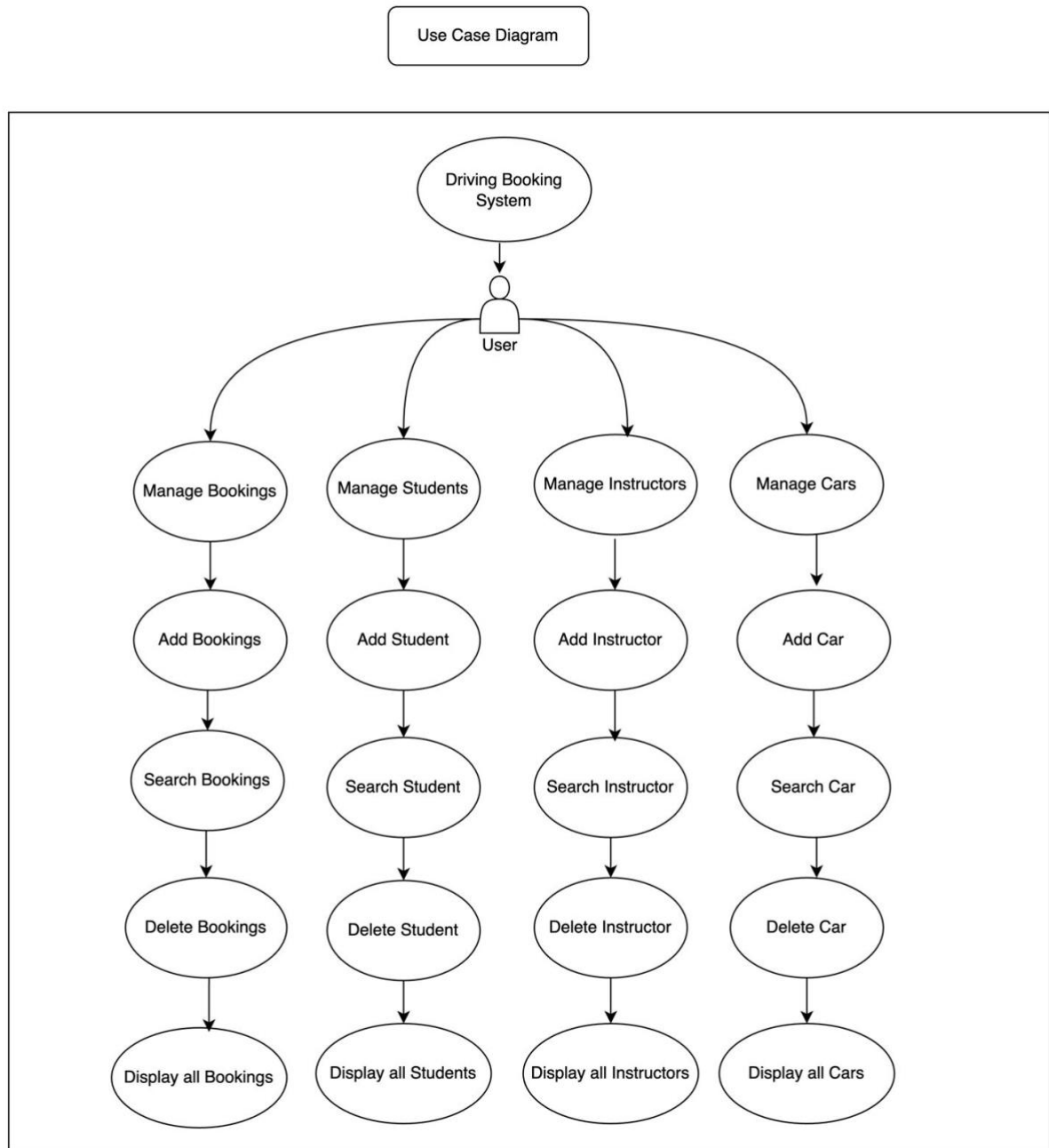
Figure 6: Car Management Flowchart

Figure 7: Use Case Diagram

Testing

The team tester performed numerous tests using MStest and comprehensive test cases across the program to detect and document bugs. When errors were discovered, the tester promptly notified the team to guarantee that the issue was addressed and any additional defects were quickly resolved.

Risk Management

Risk management was integral to the project. We encountered technical challenges, including hash collisions and SQLite locking issues, along with tight deadlines. These were addressed by modifying our hash functions, optimising database access, and working in agile sprints with daily stand-up meetings. Additionally, due to time constraints, we postponed advanced conflict resolution notifications to future iterations. By fostering open communication and rapid issue resolution via tools like WhatsApp and GitHub, we ensured that risks were mitigated with minimal impact on the project timeline.