

**Trimester March/April, 2025**

**CSE6224 SOFTWARE REQUIREMENTS ENGINEERING**

**Project Part 1**

**Topic: Campus Ride-Sharing Platform with**

**Parking System Integration**

**Requirements Elicitation Report**

|  |  |  |
| --- | --- | --- |
| Name | Student ID | Course |
| Chee Rui | 1211112287 | Bachelor of Computer Science |
| Teh Li Wei | 1211109581 | Bachelor of Computer Science |
| Sow Chien Yee | 1211210800 | Bachelor of Computer Science |
| Lai Zi Xuan | 1211109451 | Bachelor of Computer Science |

Table of Contents

[1 Introduction 3](#_Toc198048349)

[1.1 Purpose of the Document 3](#_Toc198048350)

[1.2 Problem statement 3](#_Toc198048351)

[1.3 Objectives 3](#_Toc198048352)

[1.4 Scope 3](#_Toc198048353)

[1.5 Intended Audience 4](#_Toc198048354)

[1.6 Overview 4](#_Toc198048355)

[2 Context Objects and Requirements Sources 5](#_Toc198048356)

[2.1 System Environment Overview 5](#_Toc198048357)

[2.2 Stakeholder Identification 5](#_Toc198048358)

[2.3 Context Diagram 6](#_Toc198048359)

[2.4 Sources of Requirements 6](#_Toc198048360)

[3 Requirements Elicitation Plan 7](#_Toc198048361)

[3.1 Selected Elicitation Techniques 7](#_Toc198048362)

[3.2 Justification for Using the Kano Model Elicitation Strategy Overview 7](#_Toc198048363)

[3.3 Classification of Requirements Justification for Using the Kano Model 8](#_Toc198048364)

[3.4 Classification of Requirements Using Kano Model 8](#_Toc198048365)

[4 Elicitation Execution and Findings 8](#_Toc198048366)

[4.1 Summary of Elicitation Sessions 8](#_Toc198048367)

[4.2 Categorized Requirements (Based on Kano) 8](#_Toc198048368)

[4.3 Observations and Notes 9](#_Toc198048369)

[5 Appendices 9](#_Toc198048370)

[5.1 Raw Notes or Transcripts 9](#_Toc198048371)

[5.2 Survey Results or Interview Templates 10](#_Toc198048372)

[5.3 References 10](#_Toc198048373)

1 Introduction

1.1 Purpose of the Document

The purpose of this documentation is to systematically gather, extract, organize, and present the requirements and expectations of stakeholders to ensure a clear understanding of the software’s intended functionality through various elicitation methods. It serves as a reference for both clients and developers to clarify expectations, align goals, and provide a foundation for further analysis, validation, and design.

## 1.2 Problem statement

The Multimedia University (MMU) Cyberjaya campus frequently experiences issues related to limited parking availability, illegitimate parking practices, and the absence of coordinated transportation options for students and staff. As a result, there is a need for a system that simplifies the search for available parking, enables the reporting of unauthorized usage, and provides transportation solutions to enhance campus mobility.

## 1.3 Objectives

* Identify the crucial functions required by users
* Identify the preferred behaviour and details of each function
* Identify the systems and APIs that interact with the system
* Capture both functional and non-functional requirements
* Uncover any constraints or limitations

## 1.4 Scope

The scope of this elicitation process covers the identification of user and administrative requirements for the campus ride-sharing platform and parking system at Multimedia University, Cyberjaya. It includes:

* User authentication (login via Student ID and password)
* Interactive map features for viewing and navigating parking spaces
* Viewing and reporting illegitimate parking
* Admin review and management of parking reports
* Car Pooling features

It does not include:

* Payment processing systems
* Integration with external vehicle databases
* Real-time camera surveillance or mobile application functionality

## 1.5 Intended Audience

The intended audience for this elicitation process includes all stakeholders involved in or affected by the development of the campus ride-sharing platform and parking system at Multimedia University (MMU), Cyberjaya. These include:

* Students and faculty staff as main users of the system who will interact with features such as parking space viewing, space claiming, reporting, and ride booking
* System administrators, who can view reports, view car details and overwrite parking space.
* NICE MMU, who provides the API for Student ID, password and car details.
* Software development and design team, who will use the elicited requirements to design and implement the system

1.6 Overview

Brief summary of what’s in section 2 3 4 basically the other sections (Probably leave till last)

2 Context Objects and Requirements Sources

2.1 System Environment Overview

The system will operate within the MMU Cyberjaya campus environment and will be accessible via the pre-existing MMU mobile app. It is intended to extend the app’s current functionalities and address parking-related issues while leveraging existing campus systems such as user authentication databases and car detail records.

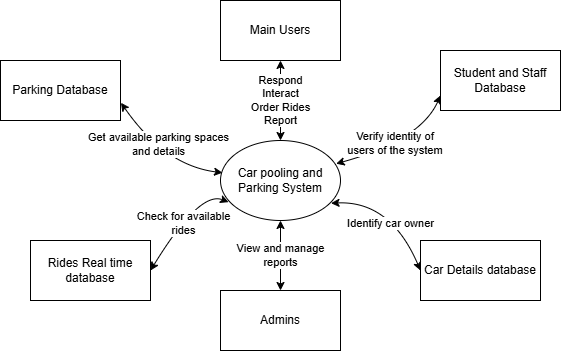
The environment includes:

* Existing Mobile App Platform: Users will be much more familiar with it and prevents the creation of another mobile app.
* User devices: Mobile phones, offering convenience for users interacting with the system while in their vehicles.
* Authentication system: Integration with MMU's student/staff ID database for login
* Car Details system: Integration with the current Car Sticker protocol, ensuring stricter and more controlled vehicle access within the campus.
* Administrator backend access: Restricted access panel for authorized personnel to view and manage parking reports

2.2 Stakeholder Identification

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Role** | **Interest / Responsibility** |
| Students | Primary users | Use the app to find parking, claim parking space, report issues, and book rides; expect convenience and speed. |
| Faculty & Staff | Secondary users | Similar to students; usage of ride function may be less or negligible |
| System Administrators | Admin users | Manage reports, review user-submitted data, and maintain system integrity. |
| University Security Staff | Admin users | Check reports, verify reports and remove resolved reports |
| University IT Department | Technical support | Provide infrastructure, handle API maintenance with existing databases (login, car info). |
| Developers | System implementers | Use elicited requirements to design, develop, and test the application. |

2.3 Context Diagram



The context diagram above shows the high-level interactions between the proposed parking and transportation system and its external entities. These entities include students and faculty/staff as users, system administrators, internal systems like the real time rides database and external systems such as MMU's authentication and car sticker databases. The diagram describes the system's boundaries and the main data flows.

2.4 Sources of Requirements

The requirements for our system are gathered from the following sources:

* End Users (Students and Staff): Provide insight into daily parking issues, feature expectations, and user experience needs.
* System Administrators: Offer requirements related to managing reports and verifying car details.
* University IT Department: Define technical constraints, API details for database, and infrastructure capabilities.
* Existing Campus Policies and Procedures: Inform functional boundaries, such as parking regulations and transport access rules.
* Elicitation Techniques: Data collected via interviews, questionnaires and prototyping with stakeholders.

3 Requirements Elicitation Plan

3.1 Selected Elicitation Techniques

Out of the many elicitation techniques, we decided to pick 3 of the most crucial to effectively gather key specifications and user expectations for the proposed system from our stakeholders.

1. Questionnaire

This technique helps to collect general opinions and preferences regarding parking and ride sharing features.

* Cost-effective and time-efficient
* Reaches a large number of respondents quickly
* Useful for identifying common issues and feature expectations

2. Interview

One-on-one interviews were conducted with selected stakeholders, including students, system administrators, and university IT staff to share thoughts and explore specific requirements in detail.

* Allows follow-up questions and clarification
* Provides detailed, context-rich insights
* Helps understand workflows, exceptions, and stakeholder goals

3. Prototyping

Figma prototypes were developed to demonstrate possible layouts and overall of the system. These were shown to users for feedback and confirmation.

* Helps validate assumptions before implementation
* Encourages stakeholders to express preferences and UI concerns
* Useful in discovering hidden usability or feature gaps

3.2 Justification for Using the Kano Model Elicitation Strategy Overview

 Summarize how requirements are gathered (e.g., interviews, surveys, task analysis).

 Describe team roles (who did what).

3.3 Classification of Requirements Justification for Using the Kano Model

Why we choose kano model preferbally bullet point form

3.4 Classification of Requirements Using Kano Model

Define the categories: Dissatisfiers, Satisfiers, Delighters   
Talk about what are they  
How are we going to sort (There’s methods)

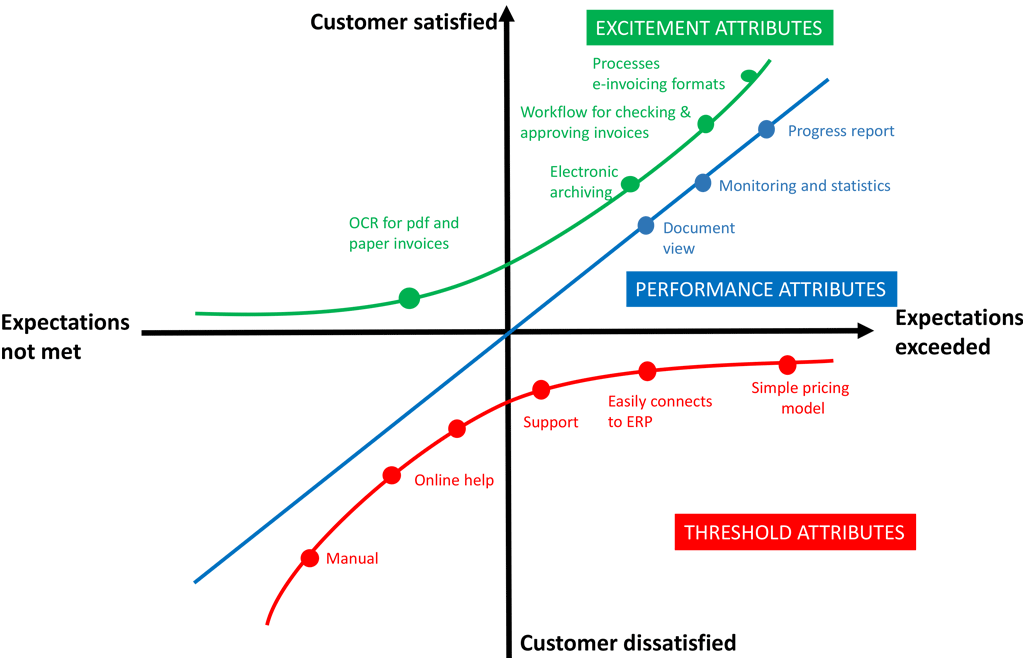
4 Elicitation Execution and Findings

4.1 Summary of Elicitation Sessions

(provides an overview or summary of the elicitation activities you’ve conducted, typically includes a summary of what was discussed, the main points or insights gathered, and any key takeaways from those sessions. It is more about describing the process, context, and the outcomes from the sessions themselves, without delving into the specifics of the requirements.)

4.2 Categorized Requirements (Based on Kano)

Define the categories: Dissatisfiers, Satisfiers, Delighters   
Sort the requirements into those categories above then create the kano model  
  
example



4.3 Observations and Notes

* General findings and interesting things noticed.
* Conflicting or ambiguous feedback.
* Anything that impacted what made it into the SRS.

5 Appendices

5.1 Raw Notes or Transcripts

fsdfsddf

5.2 Survey Results or Interview Templates

fsdfsddf

5.3 References

fsdfsddf