

Parkalot



By:

Aqib Ur Rehman

27953

Huzaifa Khan

27951

Muhammad Waqas Rathor

28550

Supervised by:

Mr. Muhammad Imran Khan

Faculty of Computing

Riphaah International University, Islamabad

Spring/Fall 2024

A Dissertation Submitted To

Faculty of Computing,

Riphah International University, Islamabad

As a Partial Fulfillment of the Requirement for the Award of the

Degree of

Bachelors of Science in Computer Science

Faculty of Computing

Riphah International University, Islamabad

Date: [date of final presentation]

Final Approval

This is to certify that we have read the report submitted by *name of student(s) (CMS #)*, for the partial fulfillment of the requirements for the degree of the Bachelors of Science in Computer Science (BSSE). It is our judgment that this report is of sufficient standard to warrant its acceptance by Riphah International University, Islamabad for the degree of Bachelors of Science in Computer Science (BSSE).

Committee:

1

Mr.Muhammad Imran Khan
(Supervisor)

2

Dr.Musharraf Ahmed
(Head of Department/chairman)

Declaration

We hereby declare that this document “[**Project Title**]” neither as a whole nor as a part has been copied out from any source. It is further declared that we have done this project with the accompanied report entirely on the basis of our personal efforts, under the proficient guidance of our teachers, especially our supervisor [**insert name of Supervisor(s)**]. If any part of the system is proved to be copied out from any source or found to be reproduction of any project from anywhere else, we shall stand by the consequences.

Aqib Ur Rehman

27953

Muhammad Huzaifa Khan

27953

Muhammad Waqas Rathor

28550

Dedication

This project is dedicated to all individuals striving to make urban living more efficient and sustainable. We extend our gratitude to our mentors, family, and friends for their unwavering support and encouragement throughout the development of Parkalot. Their guidance and belief in our vision have been instrumental in bringing this innovative parking solution to life.

Acknowledgement

First of all we are obliged to Allah Almighty the Merciful, the Beneficent and the source of all Knowledge, for granting us the courage and knowledge to complete this Project.

We would like to express our heartfelt gratitude to everyone who supported us during the development of Parkalot. Our sincere thanks to our mentors and advisors for their valuable guidance, constructive feedback, and encouragement throughout this project. We also extend our appreciation to our peers for their collaboration and insightful discussions, which enriched our understanding and inspired creative solutions.

Special thanks go to our families and friends for their unwavering support, patience, and motivation, which helped us overcome challenges and stay focused on our goals. Lastly, we are grateful for the opportunity to work on this project, which allowed us to apply our skills and contribute to addressing real-world problems in parking management

Aqib Ur Rehman
27953

Huzaifa Khan
27951

Muhammad Waqas Rathor
28550

Table of Contents

Table of Contents	vi
List of Tables	viii
List of Figures	ix
Abstract	1
Chapter 1: Introduction	2
1.1 Goals and Objectives	2
1.2 Scope of the Project	3
Chapter 2: Literature Review	1
2.1 Introduction	5
2.2 Background and Problem Elaboration	5
2.3 Detailed Literature Review	5
2.3.1 Definitions	5
2.3.2 Related Research Work 1	Error! Bookmark not defined.
2.3.3 Related Research Work 2	Error! Bookmark not defined.
2.4 Literature Review Summary Table	7
2.5 Research Gap	7
2.6 Problem Statement	7
Chapter 3: Requirements and Design	5
3.1 Requirements	9
3.1.1 Functional Requirements	9
3.1.2 Non-Functional Requirements	9
3.1.3 Hardware and Software Requirements	9
3.2 Proposed Methodology	10
3.3 System Architecture	10
3.4 Use Cases	11
3.4.1 Sample Use Case Name Here	Error! Bookmark not defined.
3.5 Database Design (<i>Optional</i>)	14
3.6 Class diagram (<i>Optional</i>)	14
3.7 Sequence diagram (<i>Optional</i>)	14
3.8 Any Other Artifact... ..	14
3.9 Graphical User Interfaces(GUI) (<i>Optional</i>)	14
Chapter 4: Implementation and Test Cases	9
4.1 Implementation	15
4.1.1 Implementation of First Component/Algorithm	15
4.2 Test case Design and description	15
4.2.1 Sample Test case No.1	15
4.2.2 Sample Test case No.2	15
4.3 Test Metrics	16
4.3.1 Sample Test case Matric.No.1	16
4.3.2 Sample Test case Metric.No.2	16
4.3.3 Sample Test case Metric.No.3	16
Chapter 5: Experimental Results and Analysis	17

Chapter 6: Conclusion and Future Directions	18
References	19
Appendix	20
Appendix A: Guidelines	20
Appendix B: Heading of Sample Appendix B	20
Formatting Guidelines	21
Chapter 1: Heading 1	21
1.1 Heading 2	21
1.1.1 Heading 3	21
Tables and Figures	22
Equations	23
Header/Footer	23
Other Formatting Guidelines	23
References	23

List of Tables

Table 1: This is Sample table caption	22
Table 2: This is Sample table caption	22

List of Figures

Figure 1: List of Styles	22
Figure 2: IEEE Reference style	23

Abstract

Parkalot is a mobile-first application designed to revolutionize parking management by bridging the gap between parking seekers and property owners offering available spaces. The application streamlines the process of finding, reserving, and managing parking spots, addressing challenges such as parking scarcity, traffic congestion, and inefficient resource utilization. It incorporates secure registration using manual identity verification through ID card images and live photo uploads, ensuring a high level of trust and transparency.

Unlike traditional systems, Parkalot adopts a simple manual payment process that eliminates the need for automated systems, making it both user-friendly and adaptable. The real-time parking spot discovery feature, coupled with a mobile-friendly interface, ensures convenience and accessibility for users. This report explores the project's development process, its goals, and the potential benefits it brings to urban mobility by offering a practical and scalable solution for parking challenges.

Chapter 1: Introduction

Urbanization has led to a surge in vehicle ownership, resulting in an increasing demand for parking spaces. The lack of efficient parking management systems has contributed to problems such as prolonged search times, traffic congestion, and underutilization of available parking spots. Parkalot aims to address these challenges by providing a mobile-based platform that simplifies the parking experience for both parking seekers and property owners.

Parkalot focuses on creating a seamless connection between users looking for parking and property owners who wish to rent their available spaces[\[1\]](#). The application's secure registration process, which involves manual ID card and live photo verification, ensures that only genuine users can access the platform. Additionally, the app offers real-time parking spot discovery, allowing users to locate and reserve spots quickly and efficiently. By integrating a straightforward manual payment system, Parkalot eliminates complexities while maintaining transparency in transactions.

This report delves into the development process of Parkalot, including its scope, objectives, and technical implementation. It also highlights the application's potential to improve urban mobility by optimizing parking space utilization and reducing traffic-related issues. By addressing these critical challenges, Parkalot aims to provide a practical, user-friendly, and impactful solution for modern urban environments.[\[2\]](#)

1.1 Goals and Objectives

Goals

1. Simplify Parking Management

The primary goal of Parkalot is to provide a seamless solution for parking management by connecting parking seekers with property owners through a mobile application[\[3\]](#). By addressing common issues such as lack of parking availability and inefficient processes, Parkalot aims to streamline the experience of finding, reserving, and utilizing parking spaces.

2. Optimize Urban Mobility

Parkalot seeks to enhance urban mobility by minimizing the time and effort spent on finding parking spaces. This goal aligns with reducing traffic congestion caused by vehicles searching for parking, ultimately contributing to more efficient traffic flow in cities.[\[4\]](#)

3. Empower Property Owners

The project aims to empower property owners by enabling them to monetize their unused parking spaces. By creating a platform for them to list and manage their spots, Parkalot provides an additional source of income while improving resource utilization.

4. Promote User Accessibility

Ensuring ease of use and accessibility for all users is a core goal of Parkalot. The application is designed to provide an intuitive and user-friendly interface, making it accessible for parking seekers and property owners, regardless of their technical expertise.

Objectives

1. Secure User Registration

Parkalot focuses on implementing a robust registration system using manual ID card and live photo verification. This ensures that only verified users can access the platform, promoting trust and security among users.

2. Efficient Parking Spot Management

The application will enable property owners to list, update, and manage their parking spots directly through the mobile app. This includes real-time updates on parking spot availability, ensuring a smooth experience for both owners and seekers.

3. Real-Time Search and Booking

Parkalot aims to provide parking seekers with a GPS-enabled search and booking system, allowing them to find and reserve parking spaces in real time. This feature will significantly reduce the hassle of searching for parking manually.[\[5\]](#)

4. Manual Payment System

A simple and transparent manual payment process will be integrated, ensuring that users can securely complete transactions without the complexity of automated systems. This approach is designed to suit the local context and preferences of users.

1.2 Scope of the Project

The scope of Parkalot is to develop a mobile application that simplifies parking management by connecting parking seekers with property owners.[\[6\]](#) The project aims to address urban parking challenges such as scarcity of spaces, inefficient utilization, and user inconvenience through the following features:

1. User Registration and Verification:

Users (parking seekers and property owners) will register by uploading their ID card images and live photos for manual identity verification, ensuring secure access to the platform.

2. Parking Spot Management:

Property owners can list, update, and manage parking spaces through the app, providing real-time updates on availability and other details.

3. Search and Booking:

Parking seekers can use GPS-enabled search functionality to locate nearby parking spaces and book them in real-time for specific durations.

4. Manual Payment System:

A straightforward manual payment process will allow users to handle transactions securely, without reliance on automated systems.

5. Mobile-Only Application:

The project exclusively focuses on a mobile app for ease of access, catering to the needs of on-the-go users.

Chapter 2: Literature Review

2.1 Introduction

The increasing challenges in urban parking management have prompted researchers to explore innovative solutions utilizing advanced technologies[7]. This chapter examines the state-of-the-art methods and systems developed to address parking issues, focusing on resource allocation, AI-powered identity verification, and user experience in location-based parking systems. By comparing these approaches with Parkalot, we aim to highlight the unique contributions and gaps in existing systems.

2.2 Background and Problem Elaboration

Urbanization has significantly increased the demand for parking, leading to issues such as traffic congestion, inefficient parking space utilization, and user frustration[8]. Existing systems have attempted to mitigate these problems through smart parking solutions, but many fail to address key challenges like secure identity verification, real-time updates, and intuitive user interfaces. Parkalot differentiates itself by combining manual ID verification, real-time spot discovery, and a mobile-centric approach, making it more accessible and practical for diverse user groups.

2.3 Detailed Literature Review

2.3.1 Definitions

Smart Parking Systems: Automated solutions leveraging technology to optimize parking space allocation and reduce search times.

AI-Powered Identity Verification: Techniques using artificial intelligence to validate user identity through facial recognition or document analysis.

Location-Based Services (LBS): Applications utilizing GPS and mapping data to deliver real-time information relevant to the user's location.

2.3.2 Smart Parking System Based on Resource Allocation

Geng and Cassandras propose a resource allocation-based smart parking system aimed at optimizing parking space utilization[9]. Their method uses algorithms to dynamically assign parking spaces based on vehicle and space availability. The system focuses on reducing search time and improving traffic flow through automated resource management.

Comparison with Parkalot:

While Geng and Cassandra's system automates resource allocation, Parkalot emphasizes a manual yet user-friendly approach to payment and verification, making it more adaptable to regions with limited technological infrastructure.

Conclusion:

This work highlights the benefits of automation in parking systems but may not address the accessibility challenges that Parkalot targets with its mobile-centric and straightforward design

2.3.3 AI-Powered Identity Verification Methods

Ma and Theera-Ampornpant explore AI-driven identity verification systems, focusing on the use of facial recognition and document validation for secure access control[\[10\]](#). Their study evaluates the accuracy and reliability of AI methods, emphasizing their potential in reducing fraudulent activities.

Comparison with Parkalot:

Parkalot integrates manual identity verification through ID card and live photo uploads, offering a simpler alternative to fully automated systems[\[11\]](#). This approach ensures security while minimizing technical dependencies.

Conclusion:

AI-powered verification systems offer superior accuracy, but Parkalot's manual verification ensures practicality and ease of use, especially in areas with limited access to AI infrastructure

2.3.4 User Experience in Location-Based Services for Parking Systems

Rastogi and Bhardwaj analyze user experience in location-based parking systems, focusing on interface design, navigation accuracy, and user satisfaction. Their work emphasizes the importance of intuitive designs and real-time information in enhancing user engagement.[\[12\]](#)

Comparison with Parkalot:

Parkalot shares a similar focus on user experience, employing a GPS-enabled interface for real-time parking spot discovery[\[13\]](#). However, Parkalot's emphasis on manual payment and verification adds a layer of simplicity that sets it apart.

Conclusion:

While location-based services enhance user experience through real-time updates, Parkalot combines this with a straightforward process to ensure accessibility for diverse users.

2.4 Literature Review Summary Table

Table 1: History of Computing Devices

No.	Name, reference	Inventor	Year	Input	Output	Description
1.	Smart Parking System Based on Resource Allocation]	Geng & Cassandras	2013	Parking space data, vehicle location	Optimized parking space assignment	Proposes a system for dynamic resource allocation to reduce search time and traffic
2.	AI-Powered Identity Verification Methods	Ma & Theera-Ampornpunt	2019	User facial images, ID documents	Verified user identity	Explores AI-driven methods for secure identity verification through facial recognition
3.	User Experience in Location-Based Services for Parking Systems	Rastogi & Bhardwaj	2020	GPS location, user preferences	Real-time parking suggestions	Analyzes user interface design and real-time navigation for location based-parking system

2.5 Research Gap

Despite significant advancements in smart parking systems, several gaps remain that limit their widespread adoption[14]. Most existing systems, such as those based on resource allocation and AI-driven identity verification, rely heavily on advanced infrastructure and automation, which may not be feasible in regions with limited technological resources. Additionally, while user experience in location-based services has been explored, many solutions lack inclusivity, focusing solely on automated systems that may exclude users who prefer or require simpler, manual processes.

Parkalot addresses these gaps by combining the efficiency of GPS-enabled parking discovery with the accessibility of manual identity verification and payment methods[15]. This ensures that users can benefit from a secure and straightforward parking solution without the need for extensive automation or complex integrations.

2.6 Problem Statement

Urbanization has exacerbated parking challenges in cities, leading to inefficient resource utilization, prolonged search times, and increased traffic congestion. While existing smart parking systems aim to resolve these issues, they often require advanced technological

Parkalot

infrastructure and fail to cater to users in regions where automation is impractical or inaccessible.[\[16\]](#)

There is a critical need for a user-centric parking solution that balances technological innovation with simplicity and adaptability. Parkalot addresses this by offering a mobile application that integrates manual identity verification, real-time parking spot discovery, and a user-friendly payment process. This project seeks to bridge the gap between advanced parking systems and the practical needs of users in diverse urban settings.

Chapter 3: Requirements and Design

Describe all modules of requirements and design in clear English text along with the necessary diagram and figures. Anyone reading your report should be able to reproduce your system/results after reading it.

For each chapter provide a paragraph of introduction and in the end a paragraph of conclusions. Make sure no heading/subheading is blank. Write text to introduce each section as well.

Introduce sub-heading as:

3.1 Requirements

3.1.1 Functional Requirements

ID	Requirments
3.1.1.1	The system shall allow users (parking seekers and property owners) to register an account
3.1.1.2	The system shall require users to upload ID cards and live photos for manual verification.
3.1.1.3	Property owners shall be able to list, update, and manage their parking spots
3.1.1.4	The system shall allow parking seekers to search for available parking spots using GPS.
3.1.1.5	The system shall enable parking seekers to book parking spots in real-time.
3.1.1.6	The system shall support manual payment processing between users.
3.1.1.7	The system shall maintain a booking history for parking seekers to view heir previous reservations.
3.1.1.8	The system shall send notifications to property owners when their parking spots are booked.

3.1.2 Non-Functional Requirements

1. Usability:

The application should have a user-friendly and intuitive interface, ensuring ease of navigation.

2. Performance

The system must respond to user queries (e.g., search and bookings) within 3 seconds.

3. Security:

Ensure secure data transmission and storage for ID cards and user information.

4. Scalability:

The app should support a growing user base without performance degradation.

5. Availability:

Ensure 99% uptime for uninterrupted access.

3.1.3 Hardware and Software Requirements

Hardware Requirements:

Mobile device (Android or iOS) for users.

Server with adequate storage and processing capabilities to host user data and app logic.

GPS-enabled devices for parking seekers.

Software Requirements:

- I. **Development Platforms:** Android Studio/Visual Studio (for mobile development).
- II. **Backend Framework:** Java, Spring Boot, Flask/Django for API development.
- III. **Database:** MySQL for storing user and parking data.
- IV. **Image Processing:** OpenCV or TensorFlow for identity verification.

3.2 Proposed Methodology

The proposed methodology involves the following phases:

1. Requirement Analysis:

Understand the needs of parking seekers and property owners, focusing on simplicity and security.

2. System Design:

Develop a user-friendly mobile application with clear workflows for registration, parking management, and booking.

3. Development

Build the app with secure manual ID verification, real-time parking search functionality, and efficient booking workflows.

4. Testing and Integration:

Perform rigorous testing to ensure functionality, security, and scalability.

5. Deployment and Feedback:

Deploy the app and collect user feedback for future improvements.

3.3 System Architecture

The system architecture of Parkalot is designed as a multi-tier architecture consisting of the following layers:

1. Presentation Layer:

The mobile app interface for users to register, search, book, and manage parking.

2. Application Layer:

Business logic, including ID verification, parking management, and booking workflows, implemented through REST APIs.

3. Data Layer:

MySQL database for storing user information, parking spot details, and booking history.

4. Hardware Layer:

GPS-enabled mobile devices and servers hosting the backend and database.

This architecture ensures scalability, security, and performance while providing a seamless user experience.

3.4 Use Cases

3.4.1 Register User

Name	Register User																				
Actors	User (Parking Seeker or Property Owner)																				
Stakeholders	User: Wants to register securely. System Administrator: Ensures only verified users access the system.																				
Pre-Conditions	User must have a valid ID card and a device with a camera.																				
Success Guarantee	User is successfully registered and can log in to the system																				
Post-Conditions	User data is securely stored, and access to system features is granted.																				
Special Requirements	None																				
Summary	The user shall provide their email and password on the login form and after successful verification, redirect the user to the home page.																				
Basic Flow																					
<table><tr><th colspan="2">Actor Action</th><th colspan="2">System Response</th></tr><tr><td>1</td><td>User opens the app and selects "Register."</td><td>2</td><td>Displays the registration form.</td></tr><tr><td>3</td><td>User fills in personal details.</td><td>4</td><td>Validates the input fields and confirms details.</td></tr><tr><td>5</td><td>User uploads an ID card and captures a live photo.</td><td>6</td><td>Compares ID card image with live photo for verification.</td></tr><tr><td>7</td><td>User submits the registration form.</td><td>8</td><td>Confirms successful verification and saves user data to the database.</td></tr></table>		Actor Action		System Response		1	User opens the app and selects "Register."	2	Displays the registration form.	3	User fills in personal details.	4	Validates the input fields and confirms details.	5	User uploads an ID card and captures a live photo.	6	Compares ID card image with live photo for verification.	7	User submits the registration form.	8	Confirms successful verification and saves user data to the database.
Actor Action		System Response																			
1	User opens the app and selects "Register."	2	Displays the registration form.																		
3	User fills in personal details.	4	Validates the input fields and confirms details.																		
5	User uploads an ID card and captures a live photo.	6	Compares ID card image with live photo for verification.																		
7	User submits the registration form.	8	Confirms successful verification and saves user data to the database.																		

9	User receives a registration confirmation	10	Redirects the user to the login page.
Alternative Flow			
6-A	System prompts the user to re-upload a valid ID.	7-A	System notifies the user and redirects to reattempt registration.

3.4.2 List Parking Spot

Name	List Parking Spot		
Actors	Property Owner		
Stakeholders	Property Owner:List parking spots. Parking Seeker: Needs accurate parking availability.		
Pre-Conditions	Property owners must have a registered account.		
Success Guarantee	Parking spot is successfully listed and visible to users.		
Post-Conditions	Parking spot details are saved and searchable.		
Special Requirements	None		
Summary	This use case outlines how property owners can add and manage parking spots for seekers to find and book		
Basic Flow			
Actor Action		System Response	
1	Property Owner logs into the app.	2	Displays the dashboard with listing options
3	Owner selects "List Parking Spot."	4	Opens the form to add parking spot details.
5	Owner enters details such as location, size, and availability	6	Validates input and saves the parking spot details.
7	Confirms listing and updates the database.	8	Confirms listing and updates the database.
9	Spot is visible to parking seekers.	10	Notifies the owner of successful listing.
Alternative Flow			
4-A	System highlights missing fields and prompts to complete the form.		

3.4.3 Search Parking spot

Name	Search parking spot
Actors	Parking seeker

Stakeholders	-Parking Seeker-: Needs a quick way to find a parking spot -Parking Owner-:Wants visibility for listed spots		
Pre-Conditions	Parking Seeker is logged into the system. GPS is enabled on the user’s device		
Post-Conditions	Parking seeker finds and selects a parking spot for booking.		
Special Requirements	None		
Summary	This use case enables user to locate and book nearby parking spaces using GPS and search filters.		
Basic Flow			
Actor Action		System Response	
1	. Parking Seeker logs in and selects "Search."	2	Displays the search interface with filters.
3	Seeker enters search criteria (e.g., location, availability)	4	Processes the search query and retrieves matching results..
5	Seeker confirms the selection.	6	Reserves the parking spot and updates the database.
Alternative Flow			
2-A	System suggests nearby options or allows the user to adjust filters.	1-A	System prompts the user to manually enter the location.

3.4.4 Manual Payment Processing

Name	Process Manual Payment		
Actors	User (Parking Seeker or Property Owner)		
Stakeholders	Parking Seeker: Needs a simple payment process. Property Owner: Expects timely payments.		
Pre-Conditions	Booking has been confirmed by the Parking Seeker.		
Post-Conditions	Payment is completed, and the booking is finalized.		
Special Requirements	None		
Summary	Parking seekers and property owners manually process payments after confirming bookings.		
Basic Flow			
Actor Action		System Response	
1	Parking Seeker confirms the booking.	2	Displays payment details for the transaction.
3	Seeker initiates manual payment	4	Sends confirmation of payment initiation to the Property Owner.
5	Property Owner confirms receipt of payment	6	Marks the booking as complete and updates the database.

Alternative Flow			
3-A	System flags the transaction as pending and notifies both parties.	4-A	System notifies the user and redirects to reattempt registration.

3.4.5 View Booking History

Name	View Booking History		
Actors	Parking Seeker		
Stakeholders	Parking Seeker: Wants to track their previous bookings.		
Pre-Conditions	User has made at least one booking.		
Post-Conditions	Booking history is displayed for the user.		
Special Requirements	None		
Summary	Parking seekers can view past bookings, including details like date, time, and location.		
Basic Flow			
Actor Action		System Response	
1	Parking Seeker logs in and selects "View Booking History."	2	Retrieves and displays a list of past bookings.
3	Seeker selects a booking for details.	4	Displays booking information such as location, date, and amount paid
Alternative Flow			
2-A	System displays a message indicating no booking history is available.		

3.5 Database Design (*Optional*)

3.6 Class Diagram (*Optional*)

3.7 Sequence diagram (*Optional*)

3.8 Any Other Artifact...

3.9 GUI Graphical User Interfaces (*Optional*)

This section should give the GUI dumps of each screen, with reference to the user. The navigation flow of each user is also required, and each GUI should mark the functionality/use case that it covers.

Chapter 4: Implementation and Test Cases

For each chapter provide a paragraph of introduction and in the end a paragraph of conclusions.

4.1 Implementation

Whatever implementation that you have done so far, please elaborate here.

Give clear details of the algorithms that were implemented along with the platform and the APIs which were used. **For FYP-I, this chapter can be changed to description of prototype developed.**

4.1.1 Implementation of First Component/Algorithm

Write implementation of first component of your system here.

4.2 Test case Design and description

This section will be added in FYP-II. Summarize the common attributes of test cases. This may include input constraints that must be true for every input in the set of associated test cases, any shared environmental needs, any shared special procedural requirements, and any shared case dependencies. The following scheme is recommended for describing test cases in detail.

4.2.1 Sample Test case No.1

<Software component Name>			
<Reference>			
Test Case ID:	<i>Reference Number</i>	Test Date:	<i>Date</i>
Test case Version:	<i>Version number</i>	Use Case Reference(s):	<i>Relation to use cases</i>
Revision History:	<i>Refer to previous test case identity (if any)</i>		
Objective	<i>Need and scope of the testing</i>		
Product/Ver/Module:	<i>Refer to overall system being built and the place of this test case in it.</i>		
Environment:	<i>Necessary and desired properties of the test environment. (hardware/software)</i>		
Assumptions:	<i>Assumptions that might affect the testing process.</i>		
Pre-Requisite:	<i>Necessary condition that needs to be fulfilled prior to the test case.</i>		
Step No.	Execution description	Procedure result	
	<i>Events being tested.</i>	<i>Mention software response.</i>	
Comments:			
<input type="checkbox"/> <i>Passed</i> <input type="checkbox"/> <i>Failed</i> <input type="checkbox"/> <i>Not Executed</i>			

4.2.2 Sample Test case No.2

.

.

.

4.3 Test Metrics

Summarize here the common ground of attributes of test case metrics.

4.3.1 Sample Test case Metric.No.1

Metric:	Purpose
Number of Test Cases:	Total number of test cases that you have developed for your system.
Number of Test Cases Passed:	The number of test cases that successfully passed
Number of Test Cases Failed:	The number of test cases that failed
Test Case Defect Density:	(No of test cases failed * 100) No of test cases executed
Test Case Effectiveness:	No of defects detected using test cases *100 Total number of defects detected
Traceability Matrix:	Traceability is the ability to determine that each feature has a source in requirements and each requirement has a corresponding implemented feature.

4.3.2 Sample Test case Metric.No.2

4.3.3 Sample Test case Metric.No.3

.

.

.

Chapter 5: Experimental Results and Analysis

This chapter will be added in FYP-II. Give proper analysis and discussion of experimental results (in plain English text) along with tables of results. **For each chapter provide a paragraph of introduction and in the end a paragraph of conclusions.**

Chapter 6: Conclusion and Future Directions

This chapter is mandatory. Give conclusions and summary of the work done. What were your findings and what were the results? Discuss in detail whether the scope of your project was entirely covered or not and whether the objectives of the project were met or not. What challenges did you face and what has been left out and why?

Sum up all the conclusions of all the chapters here to make a conclusion chapter. Do not repeat any text, just summarize it in different words.

Give recommendations for future work also. How your project can be further enhanced or improved? Future recommendations if someone wants to work on it. **For FYP-1 it is mandatory to list down a plan of the work to be done for FYP-2.**

References

List all important sources of information which have been consulted for this project

Appendix

Appendix A: Guidelines

This section should include all supporting information from the project that was not included in the body of the report. You should include surveys, complex statistical calculations, certain detailed tables and other such information in an appendix. The information presented in this section is important to support the work presented in the body of the report but would make it more difficult to read and understand if presented within the body of the report.

Cite the appendix items in the report narrative (write "see Appendix A") and organize appendices (e.g., Appendix A, Appendix B,

Any tables, figures, forms, or other materials that are not totally central to the analysis but that need to be included are placed in the Appendix.

Appendix B: Heading of Sample Appendix B

Following is a sample code with “code” style format.

```
Void SampleFunction(){  
    Print "Hello World.";  
}
```

Formatting Guidelines

This document also serves as style guide for final year project reports. In order to give a similar high-quality appearance to all final year software project reports this template uses a collection of predefined Microsoft Word formatting styles. **These styles should be used without modification or replacement.** Font in the document is *“Time New Roman”*. This template provides following styles:

- **Title** – the main title style
- **Title2** – the subtitle style
- **Body Text** – style for paragraphs
- **Caption** – the style for a figure or table caption
- **Table Description** – the style for description of table, it must be added after caption.
- **Figure Description** - the style for description of figure, it must be added after caption.
- **Code** – the style for program source code

```
int x = 10; // Writing important code
```
- **Table Header Row** – Style for the header row of table
- **Table Grid** – the style for the data rows in the tables
- **Reference** – The style for references
- **Bullets** – The style for the bullet lists
- **Numbered List**– Style for numbered lists

All Heading styles with different level numbers are listed below.

Chapter 1: Heading 1

1.1 Heading 2

1.1.1 Heading 3

1.1.1.1 Heading 4

1.1.1.1.1 Heading 5

1.1.1.1.1 Heading 6

1.1.1.1.1.1 Heading 7

1.1.1.1.1.1.1 Heading 8

1.1.1.1.1.1.1.1 Heading 9

Tables and Figures

Tables and figures should be centered horizontally. The caption button should be used to insert caption for both the figures and tables. All figures and tables must be numbered properly. Always refer to tables and figures according to their numbers. A table or figure can be cited as follows: ‘see Table1’ or ‘as shown in Table1’. The caption of table should be centered above the table and figure caption should be centered below the figure. Place the tables/figures close to their reference. Use “Table Header Row” and ‘Table Grid’ style for table’s header and data rows respectively. It is compulsory to provide brief description of table/figure after its caption. Styles for table and figure descriptions are “Table Description” and “Figure Description” respectively.

Press Ctrl+Shift+S to see list of styles mentioned above. Figure 1 shows the Apply Style window displaying the list of styles. Select any text then press Ctrl+Shift+S, the Apply Style window will show you the current style applied on that text and if required, you can change

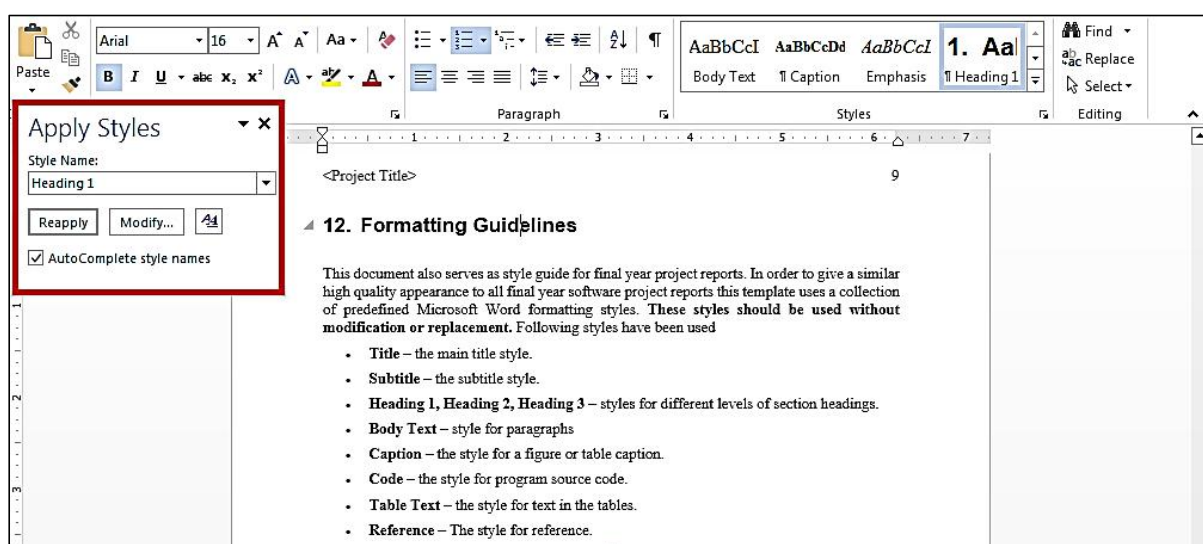


Figure 1: List of Styles

the style by selecting any other style from the “Style Name” dropdown.

This is brief description of above figure.

Table 1: This is Sample table caption

This is brief description of following Table.

Header row	Header row	Header row	Header row
Row1 col1	Row1 col2	Row1 col3	Row1 col4
Row2 col1	Row2 col2	Row2 col3	Row2 col4

Table 2: This is Sample table caption

This is brief description of following Table.

Header row	Header row	Header row	Header row
Row1 col1	Row1 col2	Row1 col3	Row1 col4
Row2 col1	Row2 col2	Row2 col3	Row2 col4

Equations

Use equation editor to write equations in this report. Use last button of the custom tool bar to invoke equation editor. Similar to tables and figures, equations should also be aligned centered horizontally. Number all equations and insert them in parenthesis. Below is a sample equation and its reference number. An equation can be referenced like this: ‘it is clear from (1)’.

$$\sum_{\forall v \in V(G)} \deg(v) = 2|E(G)| \quad (1)$$

Header/Footer

Notice the headers in this document, before Introduction (i.e. the main content of this document) page numbers are in roman numerals. The page numbers of the actual content start with Arabic numerals i.e. 1, 2, 3 and so on. All of the **odd numbered pages** contain title of your project while the **even numbered pages** contain the section heading (i.e. chapter’s name) in the headers.

Other Formatting Guidelines

- Keep 2-4 GUIs in one page. Consume as much space as possible. Do not leave most of page blank unnecessarily.
- Do not break tables (or use cases) in multiple pages unless the table is too large to fit in one page.
- Re-arrange the content i.e., text, images, and tables properly to meet above two guidelines.

References

Always refer to the source of information by inserting the reference number in square brackets like this [5]. The reference numbers can either be added at the end of the sentence or within the sentence without changing the punctuation of sentence. A reference can also be cited as follows: ‘as Ruskey [2] mentioned’. List each source only once on your reference page.

- | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>[1] B. Klaus and P. Horn, <i>Robot Vision</i>. Cambridge, MA: MIT Press, 1986.</p> <p>[2] L. Stein, “Random patterns,” in <i>Computers and You</i>, J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.</p> <p>[3] R. L. Myer, “Parametric oscillators and nonlinear materials,” in <i>Nonlinear Optics</i>, vol. 4, P. G. Harper and B. S. Wherret, Eds. San Francisco, CA: Academic, 1977, pp. 47-160.</p> <p>[4] M. Abramowitz and I. A. Stegun, Eds., <i>Handbook of Mathematical Functions</i> (Applied Mathematics Series 55). Washington, DC: NBS, 1964, pp. 32-33.</p> <p>[5] E. F. Moore, “Gedanken-experiments on sequential machines,” in <i>Automata Studies</i> (Ann. of Mathematical Studies, no. 1), C. E. Shannon and J. McCarthy, Eds. Princeton, NJ: Princeton Univ. Press, 1965, pp. 129-153.</p> <p>[6] Westinghouse Electric Corporation (Staff of Technology and Science, Aerospace Div.), <i>Integrated Electronic Systems</i>. Englewood Cliffs, NJ: Prentice-Hall, 1970.</p> <p>[7] M. Gorkii, “Optimal design,” <i>Dokl. Akad. Nauk SSSR</i>, vol. 12, pp. 111-122, 1961 (Transl.: in L. Pontryagin, Ed., <i>The Mathematical Theory of Optimal Processes</i>. New York: Interscience, 1962, ch. 2, sec. 3, pp. 127-135).</p> <p>[8] G. O. Young, “Synthetic structure of industrial plastics,” in <i>Plastics</i>, vol. 3, <i>Polymers of Hexadromicon</i>, J. Peters, Ed., 2nd ed. New York: McGraw-Hill, 1964, pp. 15-64.</p> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Figure 2: IEEE Reference style

This figure represents the styling information for adding references in IEEE format

Following is a list of sample reference for various typed of sources in IEEE format.

- [1] P.M. Morse and H. Feshback, *Methods of Theoretical Physics*. New York: McGraw Hill, 1953. **//Format for Book**
- [2] S.K. Kenue and J.F. Greenleaf, "Limited angle multifrequency diffraction tomography," *IEEE Trans. Sonics Ultrason.*, vol. SU-29, no. 6, pp. 213-217, July 1982. **//Format for Journal Article**
- [3] B. Tsikos, "Segmentation of 3-D scenes using multi-modal interaction between machine vision and programmable mechanical scene manipulation," Ph.D. dissertation, Univ. of Pennsylvania, BCE Dept., Philadelphia, 1987. [Add if applicable: University Microfilms, Inc., University of Michigan, Ann Arbor, Michigan.] **//Format for Dissertation or thesis**
- [4] R. Finkel, R. Taylor, R. Bolles, R. Paul, and J. Feldman, "An overview of AL, programming system for automation," in *Proc. Fourth Int. Joint Conf Artif. Intell.*, pp. 758-765, Sept. 3-7, 1975. **//Format for Proceedings paper**
- [5] "Technology threatens to shatter the world of college textbooks," *The Wall Street Journal*, vol 91, pp. A1, A8, June 1, 1993. **//Format for Newspaper article**
- [6] R. Cox and J. S. Turner, "Project Zeus: design of a broadband network and its application on a university campus," Washington Univ., Dept. of Comp. Sci., Technical Report WUCS-91-45, July 30, 1991. **//Format for Technical Report**
- [7] M. Janzen, *Instant Access Accounting*. Computer software. Nexus Software, Inc IBM-PC, 1993. **//Format for Software**
- [8] Fuminao Okumura and Hajime Takagi, "Maglev Guideway On the Yamanashi Test Line," <http://www.rtri.or.jp/rd/maglev2/okumura.html>, October 24, 1998. **//Format for World Wide Web (give author and title if named)**
- [9] "AT&T Supplies First CDMA Cellular System in Indonesia," <http://www.att.com/press/1095/951011.nsa.html>, Feb 5, 1996. **//Format for World Wide Web**