**Final Year Project Report**

**Project Name: Indoor Positioning System**

****

**Project Advisor: Hassan Maqbool**

**Submitted By:**

**Muhammad Nabeel (13022020154)**

**Zain-ul-abdeen (13022020172)**

**Rahman Ali (13022020076)**

**Raheel Khan (14024020023)**

**Session: 2017**

**University of Management and Technology**

**C-II Johar Town Lahore Pakistan**

**Dedication**

## 

**Final Approval**

**Panel of Examiners**

* **Head of Department**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department of Computer Science

UMT Lahore

* **Program Director ( Final Year Projects)** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department of Computer Science

UMT Lahore

* **Supervisor** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Department of Computer Science

UMT Lahore

* **Co-Supervisor** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Acknowledgment**

**Project Title**

**Objective**

**Undertaken by**

**Supervised by**

**Starting Date**

**Completion Date**

**Tools Used**

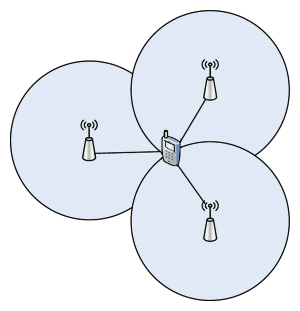
**Operating System**

**Documentation**

**Plagairism ReportAbstract**

A good navigation tool should have both a good positioning accuracy and an accessible interface. Thus, our first goal is to advance the performance of the indoor positing system on android smart phones.

The proposed indoor positioning system is based on the Beacons i.e. BLE Beacons and the android smart phones with the user friendly interface, with the accuracy of the current and destination position of the user. The perfect accuracy can be done by getting devices into range through RSSI (Received Signal Strength Indicator) and by using triangulation method (*validation of data through cross verification from two or more sources*).



The implementation of the indoor positioning system based on android has been done by using the software named as Android Studio Version 2.2.

Revision Chart

This chart contains a history of this document’s revisions. The entries below are provided solely for illustration purposes. Those entries should be deleted until the revision/s they refer to have actually been created.

The document itself should be stored in revision control, and a brief description of each version should be entered in the Revision Control System. A brief description can be repeated in this section. Revisions need not be described elsewhere in the document, unless they explain the document.

| Version | Primary Author(s) | Description of Version | Date Completed |
| --- | --- | --- | --- |
| *Draft* | TBD | Initial draft created for distribution and review comments | (To be decided) TBD |
| *Preliminary* | TBD | Second draft incorporating initial review comments, distributed for final review | TBD |
| *Final* | TBD | First complete draft, which is placed under change control | TBD |
| *Revision 1* | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| *Revision 2* | TBD | Revised draft, revised according to the change control process and maintained under change control | TBD |
| *Etc.* | TBD | TBD | TBD |

# Contents

*New paragraphs formatted as Heading 1, Heading 2, and Heading 3 will be added to the table automatically. To update this table of contents in Microsoft Word, put the cursor anywhere in the table and press F9. If you want the table to be easy to maintain, do not change it manually.*

[Contents 1](#_Toc470104823)

[Definitions and Acronyms 3](#_Toc470104824)

[List of Figures 4](#_Toc470104825)

[List of Tables 5](#_Toc470104826)

[1. Introduction 6](#_Toc470104827)

[1.1 Motivations 6](#_Toc470104828)

[1.2 Project Overview 6](#_Toc470104829)

[1.3 Problem Statement 6](#_Toc470104830)

[1.4 Objectives 6](#_Toc470104831)

[2. Domain Analysis 8](#_Toc470104855)

[2.1 Customer 8](#_Toc470104856)

[2.2 Stakeholders 8](#_Toc470104857)

[2.3 Affected Groups with social or economic impact 8](#_Toc470104858)

[2.4 Dependencies/ External Systems 8](#_Toc470104859)

[2.5 Reference Documents 8](#_Toc470104860)

[2.5.1 Related Projects 9](#_Toc470104861)

[2.5.2 Feature Comparison 9](#_Toc470104862)

[3. Requirements analysis 10](#_Toc470104865)

[3.1 Requirements 10](#_Toc470104866)

[3.2 11](#_Toc470104884)

[li 11](#_Toc470104885)

[3.2 List of Actors 11](#_Toc470104897)

[3.3 List of use cases 11](#_Toc470104898)

[3.4 System use case diagram 11](#_Toc470104899)

[3.5 Extended use cases 12](#_Toc470104900)

[3.6 User interfaces (mock screens) 13](#_Toc470104901)

[4. Data flow diagram (optional) 15](#_Toc470104902)

[4.1 Data Flow Diagram Level 0 15](#_Toc470104903)

[4.2 Data Flow Diagram Level 1 15](#_Toc470104904)

[4.3 Data Flow Diagram Level 2 15](#_Toc470104905)

[5. System Design 17](#_Toc470104908)

[5.1 System Architecture Diagram 17](#_Toc470104909)

[5.2 Class Diagram 18](#_Toc470104910)

[5.3 Collaboration Diagrams 18](#_Toc470104911)

[5.4 Other UMLs 19](#_Toc470104920)

[5.5 ERD 20](#_Toc470104921)

[5.6 Data Dictionary 20](#_Toc470104922)

[6. Implementation details 21](#_Toc470104936)

[6.1 Development Setup 21](#_Toc470104937)

[6.2 Deployment setup 21](#_Toc470104938)

[6.3 Algorithms 21](#_Toc470104939)

[6.4 Constraints 21](#_Toc470104940)

[6.4.1 Assumptions 21](#_Toc470104941)

[6.4.2 System constraints 21](#_Toc470104942)

[6.4.3 Restrictions 21](#_Toc470104943)

[6.4.4 Limitations 21](#_Toc470104944)

[7. Testing 22](#_Toc470104948)

[7.1 Extended Test Cases 22](#_Toc470104949)

[7.2 Decision Table 22](#_Toc470104950)

[7.2.1 Code snippet 22](#_Toc470104951)

[7.2.2 Decision coverage table 22](#_Toc470104952)

[7.3 Traceability Matrix 23](#_Toc470104953)

[7.3.1 RID vs UCID (requirements vs use cases) 23](#_Toc470104954)

[7.3.2 Prototypes (RID vs PID) 24](#_Toc470104955)

[7.3.3 Test Cases (RID vs TID) 24](#_Toc470104956)

[7.3.4 Coverage (UCID vs TID) 24](#_Toc470104957)

[8. Results/Output/Statistics 25](#_Toc470104958)

[8.1 %completion 25](#_Toc470104959)

[8.2 %accuracy 25](#_Toc470104960)

[8.3 %correctness 25](#_Toc470104961)

[9. Conclusion 26](#_Toc470104962)

[10. Future work 27](#_Toc470104963)

[11. Bibliography 28](#_Toc470104964)

[11.1 Books 28](#_Toc470104965)

[11.2 Journals 28](#_Toc470104966)

[11.3 Articles 28](#_Toc470104967)

[11.4 Research papers 28](#_Toc470104968)

[11.5 Other References 28](#_Toc470104969)

[12. Appendix 29](#_Toc470104970)

[12.1 Glossary of terms 29](#_Toc470104971)

[12.2 Pre-requisites 29](#_Toc470104975)

## Definitions and Acronyms

*Provide definitions or references to all the definitions of the special terms and acronyms used within this document*

e.g.

|  |  |
| --- | --- |
| **Acronym** | **Definition** |
| UMT | University of Management and Technology |
| IPS | Indoor Positioning System |
| RSSI | Received Signal Strength Indicator |

Table 1: table of acronyms and definitions

## List of Figures

New figures that are given captions will be added to the table automatically.

* **Insert caption:**
  1. select picture
  2. right click
  3. select “insert caption”
  4. under “options”, choose label as “figure”
  5. Under “caption”, an automatic insertion of “figure no” will appear. Give your figure an appropriate caption
* **Update list of figures:** To update this list in Microsoft Word, put the cursor anywhere in the table and press F9.
* **Note:**  If you want the table to be easy to maintain, do not change it manually.

Table 1: table of acronyms and definitions 3

Table 2: list of stakeholders 8

Figure 1: sample use case diagram with explanation 11

prototype 1: (P1) register a new member 14

Figure 2: System Architecture 17

## List of Tables

New Tables that are given captions will be added to the table automatically.

* **Insert caption:**
  1. select picture
  2. right click
  3. select “insert caption”
  4. under “options”, choose label as “table”
  5. Under “caption”, an automatic insertion of “table no” will appear. Give your table an appropriate caption
* **Update table:** To update this table of contents in Microsoft Word, put the cursor anywhere in the table and press F9.
* **Note:**  If you want the table to be easy to maintain, do not change it manually.

[Table 1: table of acronyms and definitions 3](#_Toc470104183)

[Table 2: list of stakeholders 8](#_Toc470104184)

# Introduction

In this upcoming eras of 20 centaury the world has becoming more and more global and smart. The concept of pervasive environment has becoming more practical and applicable which is providing easiness to users. GPS (Global positioning system) is well known system provided by Google for the users in order to locate their position but when we talk about the indoor localization it fails. Although with the passage of time buildings have been created on large scales like shopping malls, universities and large organizations in which to locate a specific location for a normal user it is quite difficult to approach a specific areas.

Indoor Positioning System (IPS) is an Android-based application which would provide indoor positioning services to users. IPS will guide users in universities to locate their class rooms, faculty offices, management offices, departments, cafeteria and other locations. Users would be able to navigate by specifying current location (origin) and destination points. This system would not only help navigate users, but also inform users about the services available at specific offices or places.

This application is personal guide for smart phone users and makes it easier for them to explore indoor spaces without getting lost. Users can track their friends and use the shortest path suggested by system to locations of their interest.

Time would be saved for the user by using IPS as it would save time for them, which would have been wasted if they were to personally explore the campus.

The IPS would provide many features, some of which are as below:

1. Accurate indoor positioning.
2. Navigate on any floor plan
3. Find staff member location
4. Share user’s current location
5. Have updated floor plans

## Motivations

Location based services are growing enabled by mobile devices with GPS. People spend most of the time indoors, rather than outdoors and they usually require assistance to guide them to their destination points to which they want to go to. The problem which arises is that there exists no wide-spread indoor positioning systems and services, and the GPS indoor location and the exact co-ordinates gets vague which makes it difficult to locate the exact positions. One way which can help to enable indoor positioning through local wireless networks is through personal communication devices.

## Project Overview

Indoor Positioning System (IPS) idea is proposed to locate objects via Bluetooth devices (Beacons) inside a building and places where most of positioning systems does not work. System will guide users to find their desired location and suggest available paths along with shortest path to destination. Current position (origin) and destination must be specified in order to track indoor locations. Registered indoor locations can be viewed on map and user position can be tracked on 2-D Map plan of building. IPS not only work on ground floor but locations can also be determined on upper levels.

System will be updated if customer wants to change something like floor plans and our goals is to provide friendly interface to user.

## Problem Statement

## As a matter of fact with the passage of time everything has becoming more and more complex. Especially small infrastructure like shopping malls, big organizations, universities etc. GPS offers good performance levels in open spaces but is not capable of locating indoor areas, because of the large attention introduced by buildings walls and ceilings therefore it cannot represents a ubiquitous localization method. So far a normal person it is really a problem to locate a specific place without knowing it. We believe that our system will provide easiness to these users having interaction with these places to locate their interests place in a very simple way.

## Objectives

The project will be accomplished by fulfilling the following objectives:

1. To improve accuracy of indoor positioning system.
2. To provide easy user interface.
3. To provide efficient utilization of resources of smart phone.
4. To update the floor plans when required.
5. Automatically switch level while using stairs or elevator.
6. To find specific departments or any faculty member in the area.

# Domain Analysis

## Stakeholders

List of all stakeholders along with their roles in making of the system e.g.

|  |  |
| --- | --- |
| **Stakeholder** | **Role in System** |
| Shopping Mall or Enterprise level | In shopping malls we can find the destination of any place or any shop, can be updated according to the floor level and our system will navigate the current location. |
| Universities | Our system will provide the destination of the departments and expansions of the departments. So our application become user friendly. |

Table 2: list of stakeholders

## Affected Groups with social or economic impact

* Enterprise Markets

🡪 The access to a specific place for user will be less complicated.

🡪 Customer rate increases or market value increase.

## Dependencies/ External Systems

IPS depends upon for its completion are:

* Beacons

Beacons are devices that broadcasts signals at a certain interval. In other words, [Beacons](https://kontakt.io/beacon-basics/what-is-a-beacon/) allow applications to recognize their location on a hyper-local scale, and sends signals to users based on the location. It is a technology that uses so called BLE beacons (or IBeacons) that are low-priced, small, have a long battery life and do not require an outward energy source. The device detects the signal from the beacon and can calculate roughly the distance to the beacon and hence estimate the location.

## Reference Documents

References, documentations and guidance from tutorials have been taken from online resources. Reference of research paper is also provided.

**Alt-Beacon library**

We have configured Alt-Beacon library for our android Beacon Transmitter app and Receiver app. This library can be configured to work to detect wide variety of beacons. For documentation and quick start guide we have used the following link.

**Reference link**

https://altbeacon.github.io/android-beacon-library/

**Android Beacon app**

To develop beacons receiver and transmitter app we have used android studio and alt beacon library. Help is taken from the following links.

**Reference links**

<http://www.software7.com/blog/creating-a-beacon-app-for-android-in-less-than-10-minutes-from-scratch/>

**Part-1**

<https://www.pubnub.com/blog/2015-04-14-building-android-beacon-android-ibeacon-tutorial-overview/>

**Part-2**

<https://www.pubnub.com/blog/2015-04-15-build-android-beacon-ibeacon-detector/>

**Part-3**

<https://www.pubnub.com/blog/2015-04-16-build-android-ibeacon-beacon-emitter/>

**Map level testing**

To test beacons transmitter and receiver on map we have used online content management system named Creator CMS. In this application we can configure beacons, set station and its range. When a receiving device entered in beacons area we can perform any action according to our need say to send discount notification.

**Reference link**

https://locatify.com/creator-cms/

### Related Projects

### Feature Comparison

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sr No. | Comparison Feature | FMS | BHMS | CIAS | Remarks |
| 1 | ABC | FMS covers the feature ABC completely as desired | BHMS does not support feature ABC | CIAS suggests that maximum efficiency can be achieved if ABC is implemented using algorithm abc. | Using the ABC feature from FMS and improving it with abc algorithm can provide maximum efficiency |

**Research paper**

Indoor localization and navigation for android platform by Juraj Benicek 2015

We have studied the research paper for triangulation and other techniques used in positioning system.

Reference link:

http://is.muni.cz/th/418105/fi\_m/dp.pdf

# Requirements analysis

## Requirements

The end user of this system will be using the application directly by installing the application on the mobile but they should be in the range of place for which this application is designed.

Following are the requirement for this application:

**System Requirements:-**

1. Android phone version greater than 4.3 +.
2. Android beacons.

## Functional Requirements

* **Current Location :-**

The user of this application will open the indoor localization application, with the help of closest android beacon device the related map for that area will be shown on the application screen and the present location of the user will be shown on the screen with a cursor pointer.

* **Destination Places:-**

On the front screen of the application there is a place for destination places to where user wants to go .A dropdown for destination will be appear that can be divided into parts depends upon the expansion of area . As the covered area is more complex the more dropdowns will be appear on the screens. If the user knows the exact Location name there will be the text field for user to enter the exact location and system will take the user to the targeted location.

* **Shortest Path:-**

Whenever the user selects the destination on the screen our application will calculate the shortest path and display on the screen.

* **Run time location indication:-**

The exact path will be shown on the screen so with the passage of time as the user moves so location indicator on the screen will also moves on the map.

## 

## Non Functional Requirements

* **Android version:-**

Minimum android version for this application should be above 4.3+.

* **Friendly Interface:-**

As this application is for public users so interface should be friendlier and less complex so that every user will feel easy to use the application.

* **Time Efficient:-**

As this application is run time application providing position run time so calculations will be more precise , so to achieve this system has to be more efficient in to achieve this android beacons will be of good quality and android version with installed application will be above 4.3+.

## Programming Languages:-

1. Java (for back end development)
2. XML (for front end interface)
3. Matlab (if necessary)

**Constraints**:-

* **System constraints**

As this application is only for those users contains android version minimum 4.3 so this application will only work for the users having latest android versions so restricted system requirements restricts this application for specific users.

* **Functional constraints**

As this is runtime application indicating the current location of user as the user moves it updates the current location of user according to his movement that’s why in order to maintain the precise location it is very difficult to maintain the exact location with precise measurements. If the speed of user increases the more chance of less precision increases. Also the user when goes out of the range there is a problem to maintain the location of user.

## List of Actors

Simple public users will directly interact with the application.

## List of use cases

Check beacons area range: checks whether the user is in the range of beacons or not.

Current location: shows the current location of the user on the screen map.

Destination location: destination location where user wants to go is entered into test area directly

Destination Dropdowns: for a normal user if he does not know the exact location of the destination so sub divided multiple dropdowns will appear on the screen.

Calculate Shortest Path: calculates shortest path from current location to destination points

Shows navigation:-

Shows graphical navigation to destination with cursor pointer.

## System use case diagram

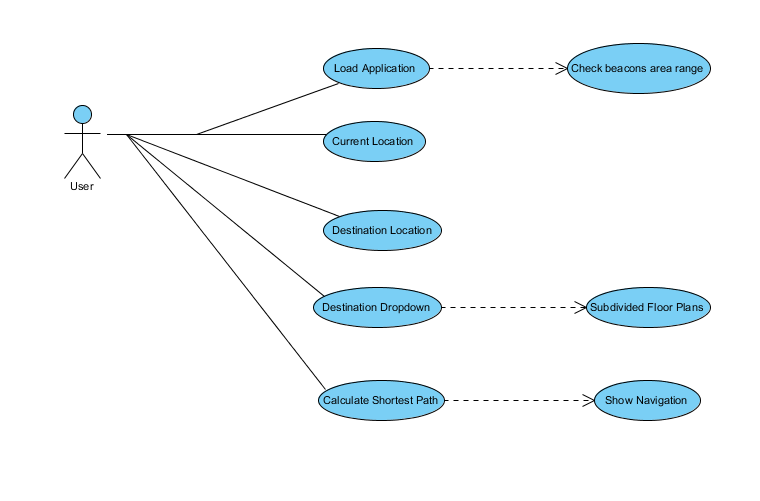


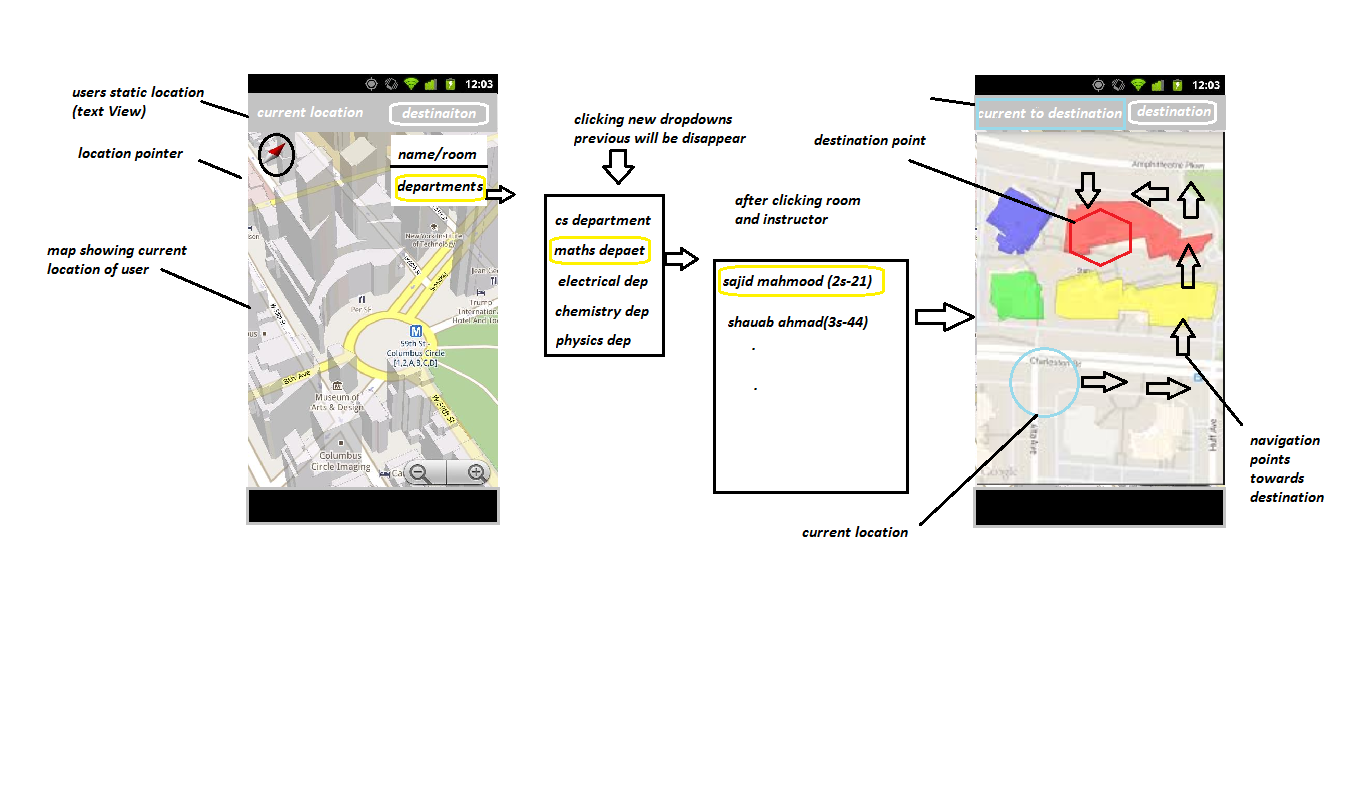
Figure 1: sample use case diagram with explanation

## Extended use cases

Every use case form the list must be elaborated here. E.g

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Use Case ID:** | Enter a unique numeric identifier for the Use Case. e.g. UC-1.2.1 | | | |
| **Use Case Name:** | Enter a short name for the Use Case using an active verb phrase. e.g. Withdraw Cash | | | |
| **Created By:** |  | | **Last Updated By:** |  |
| **Date Created:** |  | | **Last Revision Date:** |  |
| **Actors:** | | End users of this application are the main actors of this application as for the administrator there are no operator functions In this application so there is direct collaboration of the users to the application. | | |
| **Description:** | | [Provide a brief description of the reason for and outcome of this use case.] | | |
| **Trigger:** | | [Identify the event that initiates the use case. This could be an external business event or system event that causes the use case to begin, or it could be the first step in the normal flow.] | | |
| **Preconditions:** | | [List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each pre-condition. e.g.   1. Customer has active deposit account with ATM privileges 2. Customer has an activated ATM card.] | | |
| **Post conditions:** | | [Describe the state of the system at the conclusion of the use case execution. Should include both *minimal guarantees* (what must happen even if the actor’s goal is not achieved) and the *success guarantees* (what happens when the actor’s goal is achieved. Number each post-condition. e.g.   1. Customer receives cash 2. Customer account balance is reduced by the amount of the withdrawal and transaction fees] | | |
| **Normal Flow:** | | [Provide a detailed description of the user actions and system responses that will take place during execution of the use case under **normal, expected** conditions. This dialog sequence will ultimately lead to accomplishing the goal stated in the use case name and description.   1. Customer inserts ATM card 2. Customer enters PIN 3. System prompts customer to enter language performance English or Spanish 4. System validates if customer is in the bank network 5. System prompts user to select transaction type 6. Customer selects Withdrawal From Checking 7. System prompts user to enter withdrawal amount 8. … 9. System ejects ATM card] | | |
| **Alternative Flows:**  **[Alternative Flow 1 – Not in Network]** | | [Document **legitimate** branches from the main flow to handle special conditions (also known as extensions). For each alternative flow reference the branching step number of the normal flow and the condition which must be true in order for this extension to be executed. e.g. Alternative flows in the *Withdraw Cash* transaction:  4a. In step 4 of the normal flow, if the customer is not in the bank network   1. System will prompt customer to accept network fee 2. Customer accepts 3. Use Case resumes on step 5   4b. In step 4 of the normal flow, if the customer is not in the bank network   1. System will prompt customer to accept network fee 2. Customer declines 3. Transaction is terminated 4. Use Case resumes on step 9 of normal flow   Note: Insert a new row for each distinctive alternative flow. ] | | |
| **Exceptions:** | | [Describe any anticipated **error conditions** that could occur during execution of the use case, and define how the system is to respond to those conditions.  e.g. Exceptions to the Withdraw Case transaction  2a. In step 2 of the normal flow, if the customer enters and invalid PIN   1. Transaction is disapproved 2. Message to customer to re-enter PIN 3. Customer enters correct PIN 4. Use Case resumes on step 3 of normal flow] | | |
| **Includes:** | | [List any other use cases that are included (“called”) by this use case. Common functionality that appears in multiple use cases can be split out into a separate use case that is included by the ones that need that common functionality. e.g. steps 1-4 in the normal flow would be required for all types of ATM transactions- a Use Case could be written for these steps and “included” in all ATM Use Cases.] | | |
| **Frequency of Use:** | | [How often will this Use Case be executed. This information is primarily useful for designers. e.g. enter values such as 50 per hour, 200 per day, once a week, once a year, on demand etc.] | | |
| **Special Requirements:** | | [Identify any additional requirements, such as nonfunctional requirements, for the use case that may need to be addressed during design or implementation. These may include performance requirements or other quality attributes.] | | |
| **Assumptions:** | | [List any assumptions that were made in the analysis that led to accepting this use case into the product description and writing the use case description.  e.g. For the *Withdraw Cash* Use Case, an assumption could be:  The Bank Customer understands either English or Spanish language.] | | |
| **Notes and Issues:** | | [List any additional comments about this use case or any remaining open issues or TBDs (To Be Determined) that must be resolved. e.g.   1. What is the maximum size of the that a use can have?] | | |

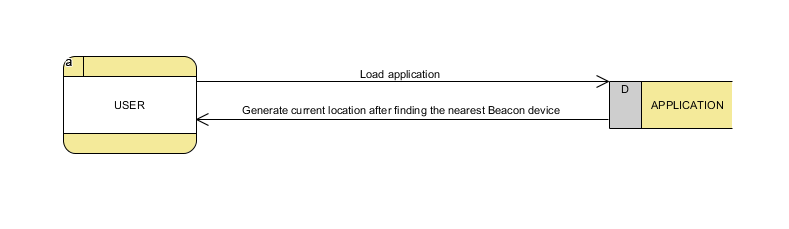
## User interfaces (mock screens)



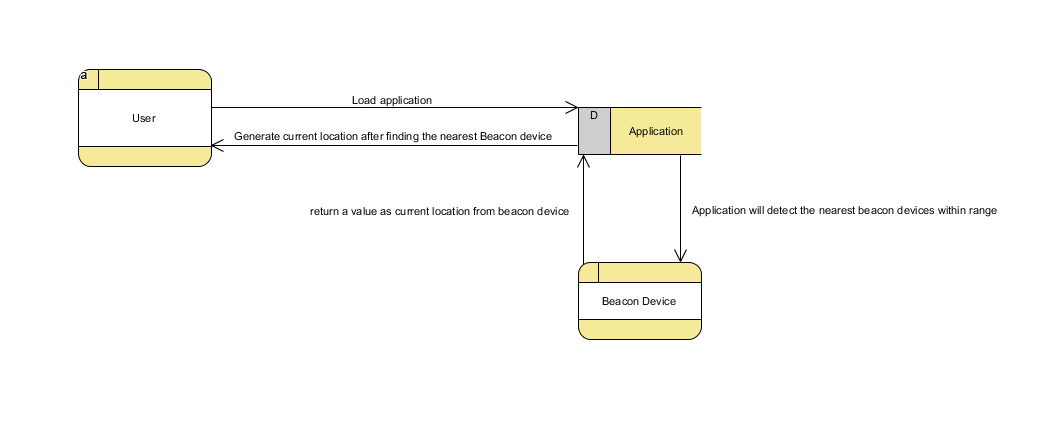
Prototype 1: (P1) register a new member

# Data flow diagram (optional)

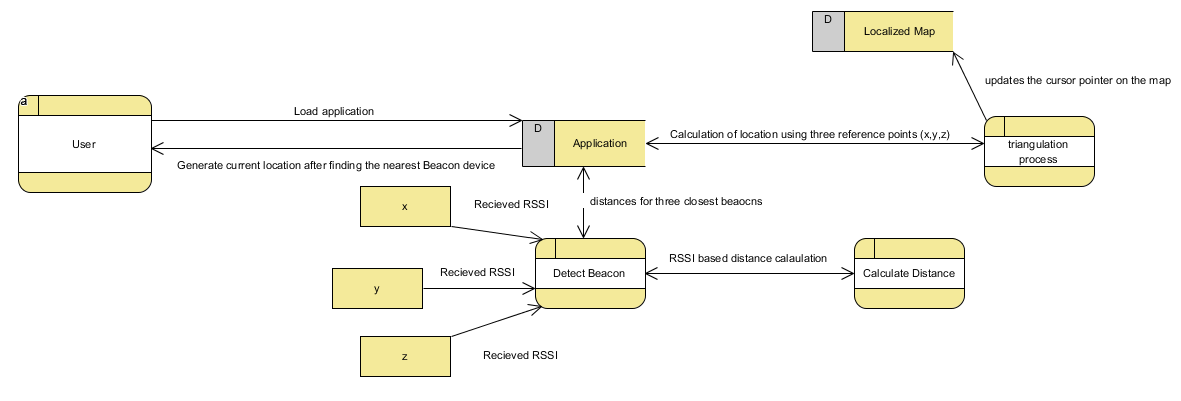
## Data Flow Diagram Level 0



## Data Flow Diagram Level 1



## Data Flow Diagram Level 2



# System Design

Describe the system architecture, or simply provide the architecture diagram. For School system it may include web based front end, webserve , database etc. Don’t worry too much about it just give a simple diagram of a typical web based project.

## System Architecture Diagram

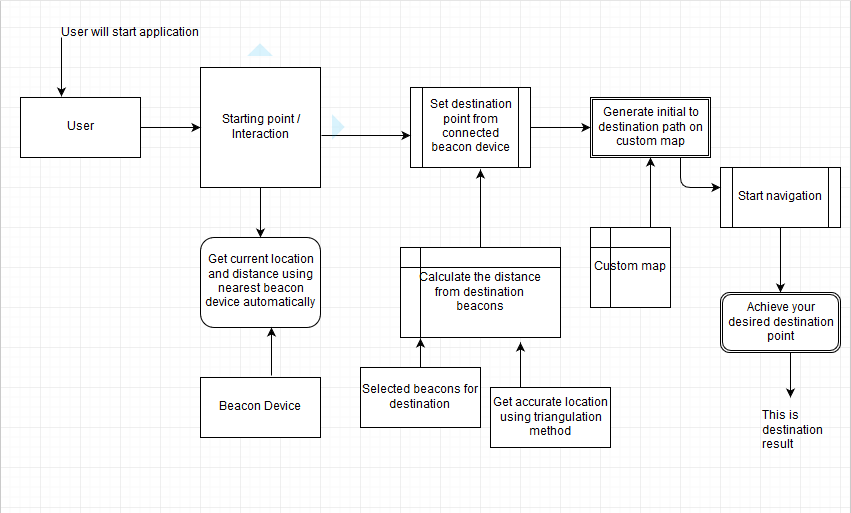
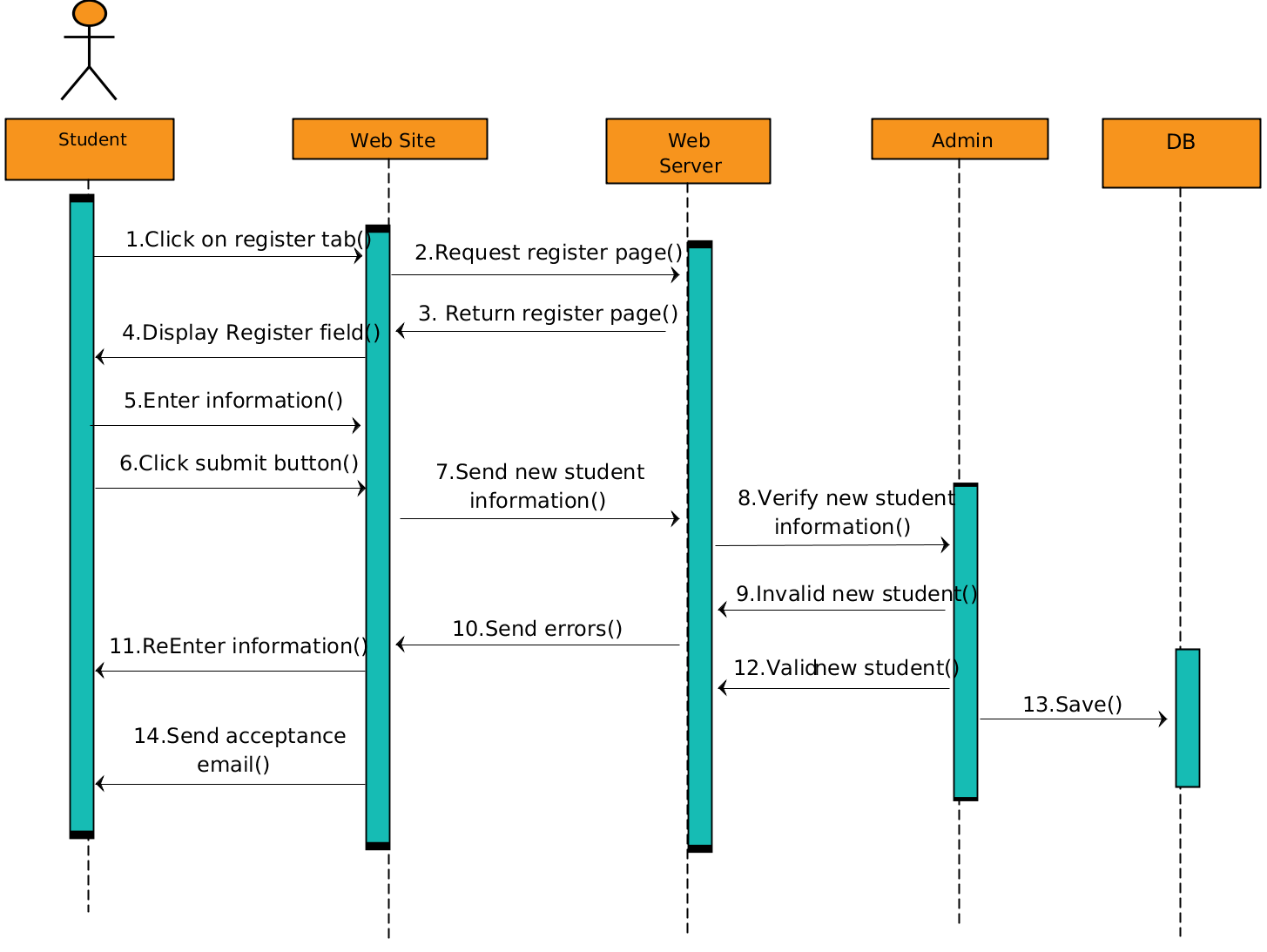


Figure 2: System Architecture

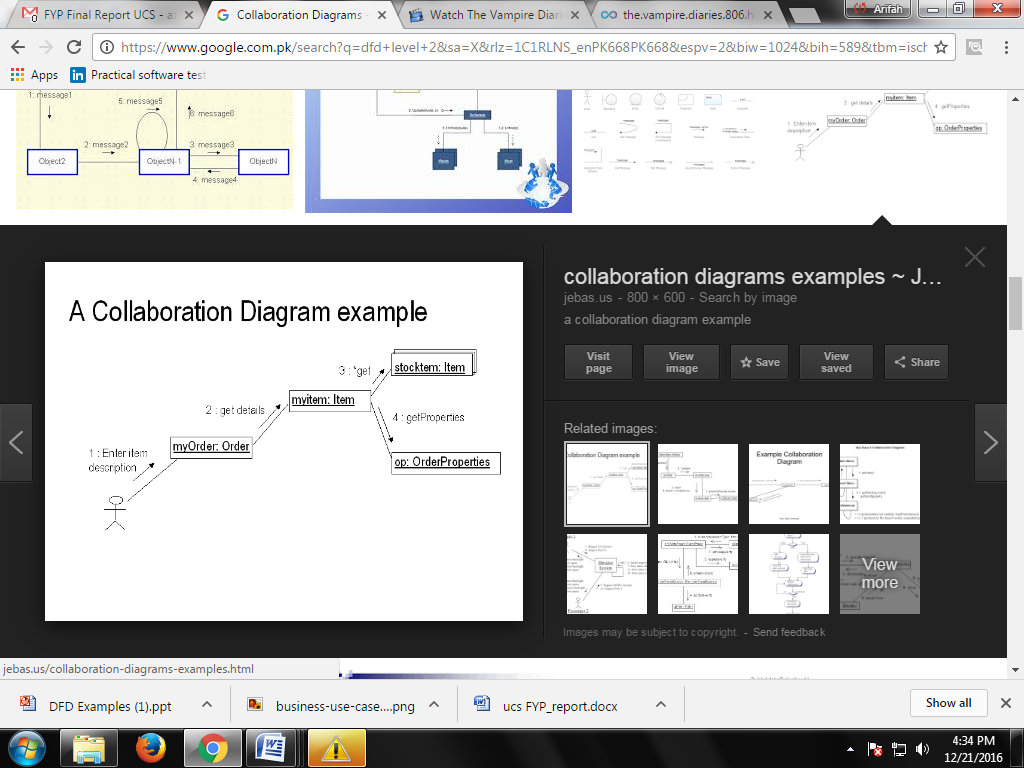
## Class Diagram

## class-example-online-shopping-domain

## Sequence Diagrams



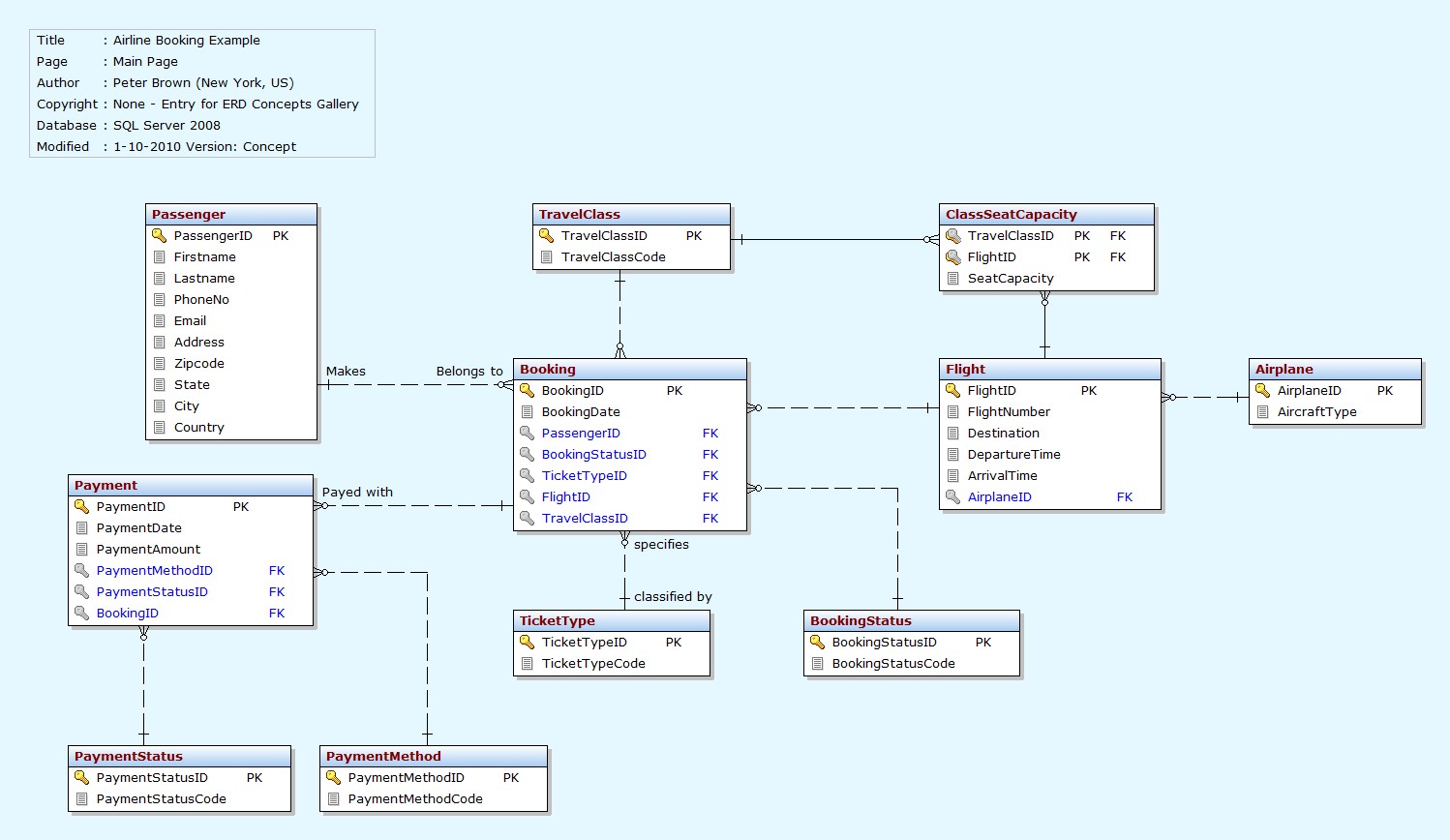
## Collaboration Diagrams



## Other UMLs

This is optional. You may include any other UML to support your system.

## ERD



## Data Dictionary

This section may be used to provide the details of interface elements that are present on the screenshots.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Element Name | Type | Validation | Mandatory | Remarks |
|  |  |  |  |  |

# Implementation details

## Development Setup

List your tools and technologies and their role in development.

## Deployment setup

How and where was your software deployed? Did you face any problems, how did you overcome these problems.

## Algorithms

Entire code of software is not required. Just highlight your important (user defined/ improved) algorithms.

## Constraints

### Assumptions

Things we assume will be true.

e.g.:

* *We will receive all necessary technical support from the engineers at cMeRun, Select and Mellon Bank to help design the interfaces between their systems and enGyro.*
* *All database maintenance will be handled by the client.*
* *There will be no real-time interfacing with any accounting systems.*

### System constraints

 A constraint specifies how the system must operate or how it must be built

### Restrictions

Constraints applied on the system by the client

### Limitations

Services your software is unable to provide

# Testing

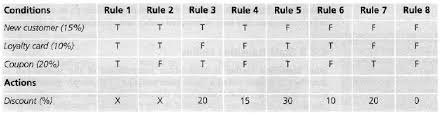
## Extended Test Cases

## 

## Decision Table

### Code snippet

### Decision coverage table



## Traceability Matrix

### RID vs UCID (requirements vs use cases)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **UCID/RID** | **R**  **1** | **R**  **2** | **R**  **3** | **R**  **4** | **R**  **5** | **R**  **6** | **R**  **7** | **R**  **8** | **R**  **9** | **R**  **10** | **R**  **11** | **R**  **12** | **R**  **13** | **R**  **14** | **R**  **15** | **R**  **16** | **R**  **17** | **R**  **18** | **R**  **19** | **R**  **20** | **R**  **21** |
| UC 1 | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 2 |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 3 | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |
| UC 4 | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |
| UC 5 | ✓ | ✓ | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 6 | ✓ | ✓ |  | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 7 | ✓ | ✓ | ✓ |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 8 | ✓ | ✓ |  | ✓ |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |
| UC 9 | ✓ | ✓ | ✓ |  | ✓ |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |  |
| UC 10 | ✓ | ✓ |  | ✓ |  | ✓ |  |  |  |  |  | ✓ |  |  |  |  |  |  |  |  |  |
| UC 11 | ✓ | ✓ | ✓ |  | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 12 | ✓ | ✓ |  | ✓ |  | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 19 | ✓ | ✓ | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |  |
| UC 20 | ✓ | ✓ |  | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |  |
| UC 21 | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 22 | ✓ | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC 23 | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |  |
| UC 24 | ✓ | ✓ |  | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |  |
| UC 25 | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |  |
| UC 26 | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |  |
| UC 27 | ✓ | ✓ | ✓ | ✓ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ✓ |

### Prototypes (RID vs PID)

### Test Cases (RID vs TID)

### Coverage (UCID vs TID)

# Results/Output/Statistics

## %completion

Use the matrix & values from 7.3.1 to show that all requirements are being fulfilled.

## %accuracy

Use the matrix & values from 7.3.3 to show that all requirements have been implemented correctly.

## %correctness

Use the matrix & values from 7.3.4 to show that all requirements have been tested to be conforming to requirements.

# Conclusion

# Future work

# Bibliography

Use IEEE or ACM format for citations

## Books

## Journals

## Articles

## Research papers

## Other References

# Appendix

## Glossary of terms

## Pre-requisites

Must use contents of development/ deployment setup & external system dependencies