**A Community for gathering people whose destination is same**

**Description**

When you travel with little amount of people, you are hesitant to take a taxi or other transport. It’s not a big problem that you are skillful traveler or the rich, but most of people feel more comfortable if they find someone move with. With your companion, you will not only save money, but also be able to start your travel more easily. Also, “sharing” is becoming the trend of era. It is not just sharing the memory or information. It can divide money and skill.

The objective of this project is to make travel more comfortable and reduce the travel expenses. Unlike the travel mate matching applications that have been released before, it does not pursue emotional exchange. It is because some security issues are important these days. It connects all the routes and saves it to protect people from the criminal by getting the information from airlines, some accommodations and gathering the location information of users only until they arrive to destination.

Users can meet and make an appointment in this app. They can use it as a messenger app, and for the safety, it never provide each others’ important information. There are options for meeting with same sex and the number of companion.

**Requirements Engineering**

**Feasibility Study**

1. Identify at least 4 examples of travel community apps in the marketplace. Briefly describe each of them. Reference the URL for each. (in Windows, Android, iOS, …)? *(min 100 words)*

1 serentrip :

<https://play.google.com/store/apps/details?id=cool.likecrazy.at&hl=en>

2 tripgrida :

<https://www.tripgrida.com/>

3 tinder : Its main purpose is not for trip, but for meeting. It has the similar function.

<https://play.google.com/store/apps/details?id=com.tinder&hl=en>

1. Identify the main the main system features and services provided in an travel community apps. Consider the existing systems and the services they provide to users.

[Reference and copy the URLs] *(min 200 words)*

1. Describe a new type of basic e-Learning system and how it might operate. Consider existing systems that provide services to users and lecturers. Use the internet to identify
2. In what ways do students currently learn to program? *(min 100 words)*
3. Who are the stakeholders? Would this app affect them positively or negatively? *(min 30 words)*
4. What other research would be necessary to ascertain feasibility e.g., ownership of smartphones/tablets, e-learning …? *(min 100 words)*
5. Make an initial list of **functional** and **non-functional** requirements.

**Requirements Elicitation**

1. Could ethnographic methods be used in this case study? If so, in what way? *(min 30 words)*
2. Identify a significant stakeholder(s), which will be **interview**ed to get more information on the intended product. Justify your choice of stakeholder(s). Do up an interview plan and pre-prepare approximately 10 questions.
3. Identify a significant group of stakeholders, which will receive **questionnaires**. Justify your choice of stakeholders. The questionnaire that you create should have approximately 10 questions.

# Requirements Analysis

1. Use the use case template to analyse the proposed system

Draw an initial *use-case diagram* with supporting scenario description for this app (possibly using *StarUML* for the diagram). The first iteration of the use-case diagram can consist of a single overall use case with supporting main flow and 2 or 3 alternative flows.

System Name

(Automated Library System)

Actor

(Student)

The use case description is developed from analysing the description of the use case. This is the statement of the goal of the use case.

For the first iteration this will be a description of the how the system operates.

Use Cases focus on functional requirements and specific system behaviour.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USE CASE** | | <number> | <Name of Use Case>  <the name is the goal as a short active verb phrase> | |
| **Description of Goal in Context** | | <a longer statement of the goal for this Use Case in context > | | |
| **Preconditions** | | <what we expect is already the state of the system>  <list> | | |
| **Post Conditions, Success End Condition** | | <the state of the system upon successful completion> | | |
| **DESCRIPTION** | | < The use case description is a paragraph identifying behaviour, it comes from the requirements gathering> | | |
| **Main Flow** | | | | |
| **Step** | **Action** | | | **Alternate** |
| n.1 | <put here the steps of the scenario from trigger to goal delivery, and any clean-up after> | | |  |
| n.2 | **<…>** | | |  |
| n.3 | **<…>** | | |  |
| n.4 | **<…>** | | |  |
| n.5 | **<…>** | | |  |
| n… | **<…>** | | |  |
|  | | | | |
| **EXCEPTIONS or ERROR Flow Description** | | | | |
| **Step** | **Branching Action**  < Exception number m of Use Case n> | | | **Alternate** |
| n.m.1 | < condition causing exception>  <Action, steps of scenario to goal delivery> | | |  |
| n.m.2 | < condition causing exception>  <Action, steps of scenario to goal delivery> | | |  |
| n.m.3 | < condition causing exception>  <Action, steps of scenario to goal delivery> | | |  |
| n.m.4 | < condition causing exception>  <Action, steps of scenario to goal delivery> | | |  |
|  | | | | |
| **ALTERNATIVE or VARIATION Flow Description** | | | | |
| **Step** | **Branching Action** | | | **Alternate** |
| n.m.1 | <condition causing alternative>  <list of variation>  <Action, steps of scenario to goal delivery> | | |  |
| n.m.2 | <Action, steps of scenario to goal delivery> | | |  |

Non-functional requirements, management issues and decisions required to be made, can be identified in the following table

|  |  |  |
| --- | --- | --- |
| **RELATED INFORMATION** | Use Case: <number> | <Use case name> |
| **Priority:** | <how critical to your system/organization> | |
| **Performance** | <the amount of time this use case should take> | |
| **Frequency** | <how often it is expected to happen> | |
| **Channels to actors** | <e.g. interactive, static files, database, timeouts> | |
| **OPEN ISSUES** | <list of issues awaiting decision affecting this use case> | |
| **Due Date** | <date or release needed> | |

1. Draw a second iteration in a separate word report consisting of 4 or 5 use cases. Each use case requires a use case narrative describing the scenario analysis. Each use case should have2 or 3 exception or alternative flows.

**Requirements Specification**

1. From the requirements analysis identified using the use Case scenario analysis identify key functional requirements.

Fill up the following matrix with *functional* requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req ID** | **Name of Req** | **Description** | **Priority** | **User Contact** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**System Modeling**

1. Identify possible *actors* in this system?
2. List the possible *use-cases* in this system?

# Verification and Validation of Requirements

1. Test Case Planning.

Put one sample test case in this section.

The additional test cases are to be in a separate test case word document.

Identify the User Acceptance Testing requirements for the use cases identified in Iteration 2 of the Use Case Analysis

Use the test case template to create initial Use Acceptance Test plans that will permit users and developers to agree the system will have been developed as specified by the requirements

Consider the test plan as a user guide or user manual for non-technical novice users of the system

|  |
| --- |
| **Test Case Number:** |
| **Test Case Name:** |
| Related Use Case  Name:  Number: |
| **Purpose:** |
| **Procedure Steps:** (Guided by Main flow or other flows of Use case) |
| **Expected Results:** |

# Completing the Feasibility Study

1. Before you do the final submission of the feasibility report, review and update your previous submissions.
2. Modify the requirements Specification list to high level (Abstract) core system features

**Update Use-Case Model and Requirement Specification**

1. Consider the Use case Model to be sure key functionality has been addressed in the analysis and modelling process.
   1. Do any of your use-cases need to be broken down further i.e., is there is too much functionality in one use-case?
   2. Update the Use Case Model as necessary.
   3. Update your RS with abstracted requirements as necessary.
   4. In your RS, put in a new column which shows what use-cases are related to which requirement.

**Prototype**

1. Sketch the home page/starting page of the app. You should take a photo of it and insert the photo into the document.

**Additional Functional & Non-Functional Test-Cases**

1. Write three additional test-case (using the test-case template) for each of three abstracted **high priority** *functional* requirements (one test-case per requirement/use case).
2. Write two test-case (using the test-case template) for each of the two **most important** *non-functional* requirements (one test-case per requirement).