[Project Overview 1](#_Toc494181933)

[Functional Features 2](#_Toc494181934)

[USE Case Diagram (To map out user stories) 2](#_Toc494181935)

[Clickable Wireframe 2](#_Toc494181936)

[ERD or UML (Database) Class Diagram 2](#_Toc494181937)

[Github 3](#_Toc494181938)

[Presentation October 3, 1:00PM 3](#_Toc494181939)

[Deductions 3](#_Toc494181940)

|  |  |
| --- | --- |
|  |  |
| Due Date: | October 13, 12PM |
| Percentage of module mark: | 55% |
| Submission Details: | Have one person on your team submit a text file that has a hyperlink to your wireframe which is to be hosted online. (Hosting details will be provided by next day.)  On the front page of your wire frame, you are to have a link to **one** word document that includes:   * Your requirements listing as described in the document. * A listing of all persons on your team. * A hyperlink to your GitHub repository. * Your use case diagram. * Your ERD / UML class (database) diagram.   Marks will be deducted for additional documents submitted. |
| Late Penalty: | 20% deducted each day this assignment is late. |

## Project Overview

In teams, your job is to design a system that will do something professionally well enough to earn sustainable revenue. As examples, you might design a system to offer a service such as event ticket sales, hotel bookings, or something else. Your design does not need to go to production but make the effort to keep it feasible and realistically scoped.

Note: Please do not design a “career posting/search, online grocery or food delivery, or job/portfolio aid” tool. Revenue generation cannot be based on ad revenue.

Teams:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Armaan  Michael  Cha  Karl  Jan | Daniel  Lin  Ben  Clayton | Edgar  Tom  Vittorio  Arelene | Khoa  Will  Duncan  Kevin | Jerry  Metkel  Manveer  Liam |

## Functional Features

**10 marks**

On one page describe the key items that make your application fully functional. Use one full page to show enough detail about the necessary features but do not exceed this limit. For your application to be successful, list the functional features under essential, important, and extra categories.

## USE Case Diagram (To map out user stories)

Provide a use case diagram to show the system goals, the sub-systems, and actors involved for functional features only. The screenshot of this diagram must fit on one page and it must be readable. Use either draw.io or visual paradigm community edition.

**15 marks**

## Clickable Wireframe

Design the web pages using static HTML. Keep it basic – **NO LIVE DATA** is permitted. Do not waste too much time making graphics, complex CSS, or JavaScript in this phase. You can embed screenshots in your clickable wireframes – keep it simple to get it done on time and within class hours.

I will discuss GitHub use and hosting options on day 4.

**25 marks**

## ERD or UML (Database) Class Diagram

Develop a relationally sound ERD or UML database diagram to model your tables, columns, and their associations. Use either draw.io , or visual paradigm community edition.

* Properly show all primary and foreign keys.
* Show proper multiplicity or cardinality.
* Show data types in your diagram that are suitable in size and format for each column.
* Use a uniform naming standard.
* Arrange your tables so they all fit on one page and so the diagram is readable on one page once the diagram is complete. Take one or more screenshot of this diagram and include it in the one word document that you submit. However you present it, make it look professional.

**20 marks**

## Github

Manage your wireframe developing using one GitHub repository. Please help each other to become familiar with GitHub. I will post information on hosting options on day 4 so you can try it then. **5 marks**

## Presentation October 3, 1:00PM

Details about the presentation grading scheme will be announced closer to the presentation date. Your team is to show off your project. The presentation will not be for marks.

## Deductions

Marks will be deducted for the following:

* Models that show anything other than essential requirements.
* Inconsistent Designs. For example, the data shown in the UML class diagram is not the same as the data in the online forms or reports.
* Hard-to-read documentation.
* Un-cropped and improperly sized images which are hard to read.
* Unprofessional appearance of documentation.
* Documentation that is submitted in a format that is different than requested.