**Software Requirements Specification (SRS) Document**

|  |  |
| --- | --- |
|  | <Your Team Number and Team Members in Black> |

Note: This is a “living document”, meaning its content will change with the implementation of the project. Use it to capture key project requirements and make sure that your product features match the requirements exactly – if you wish to add any features, they must be added first to the requirements. Changes in the document must be approved by the customer (mentor) and the instructor or TA Mentor.

*Remove this text and the descriptive paragraphs in each section stating what to do before you turn it in*. **(i.e. remove all text in blue.)**

# Brief problem statement 2-5 lines describing the problem being addressed. Note that even if you are simply restating what is already in the project description, you must rephrase it in your words. This gives an opportunity for the customer to identify and provide feedback on differences in interpretation, if any.

# Users profile Identify who will be using the product, in what mode, and their profile in terms of familiarity with using computers and such software.

# Project Modules Identify the high-level modules in your project. You will use these modules to classify and organize all your requirements. An example of a module would be a “User Management” module which handles user accounts and information. Provide a short description of what each module is, and what aspects of your product fall into each module.

**Feature requirements (described using use cases)**

For each Module described above:

1. List out submodules to further compartmentalize features (some modules may not have any submodules)
2. List out the feature requirements for the corresponding Module/Submodule.

When listing out features, maintain them as a numbered list of features of the product to be built. For each feature, provide a description (2-3 sentences) so you know what to build and so you can write a test case to demonstrate that your system provides that feature. For each feature, identify the release in which it will be implemented: R1 or R2 along with 2-3 user stories. A user story is a short description of how a user might interact with or use a feature. This can be as part of a larger pipeline or by directly interacting with the feature.

Please do be descriptive when listing out features and make sure that the feature list completely describes your system. Obvious requirements like “User Log In” must also be mentioned, even if you already have a requirement for “User Registration”.

# Feature requirements (described using use cases)

This is a numbered list of use cases that are the features of the system to be implemented. Each use case is an operation that the user can perform on/with the system. For each use case, provide a description (2-3 sentences) so you know what to build and so you can write a test case to demonstrate that your system provides that feature. For each use case, identify the release in which it will be implemented: R1 or R2. Please add or delete table rows if needed.

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **User Case Name** | **Description** | **Release** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Use Case Diagram**

Draw the UML use case diagram for the system. Make sure the use cases shown in the diagram correspond to the use cases described in the previous section.