**Product Requirements**

|  |  |
| --- | --- |
| **Team** | ☺ |

# Brief problem statement

This should be a 1-2 paragraphs describing what the motivation is for the project. What is the problem to be solved?

# System requirements

What system configuration needs to run your proposed system (including anything third party that is needed to run your system).

# Users profile

Who is the system intended for? What characteristics should the users have (this can also be a range of things such as reading level, etc.).U

# Feature requirements (user stories)

List the Priority as 1 (High Priority - Critical) to 3 (Low Priority – Would be nice if we have time)

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **User Story Name** | **Description** | **Priority** |
| 1 | Today view (sample) | Select "today" and view today's appointments. When the today button is pressed, the current day should be selected and the appts should show in the panel on the right with all the times laid out. |  |
| 2 | Delete appointment (sample) | Delete a specified appointment. when selected an option in the file menu should be able to delete an existing appt, and also confirm that they are deleting the appt. |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Use Cases for the 2-3 Most Critical Tasks

# Use Case Definition (NOTE: The following text is reference as you fill out the table for each use case; delete this instructional text before submitting this document)

## Actor

An actor is a person or other entity external to the software system being specified who interacts with the system and performs use cases to accomplish tasks. Different actors often correspond to different user classes, or roles, identified from the customer community that will use the product. Name the actor that will be initiating this use case and any other actors who will participate in completing the use case.

## Description

Provide a brief description of the reason for and outcome of this use case, or a high-level description of the sequence of actions and the outcome of executing the use case.

## Preconditions

List any activities that must take place, or any conditions that must be true, before the use case can be started. Number each precondition. Examples:

1. User’s identity has been authenticated.
2. User’s computer has sufficient free memory available to launch task.

## Postconditions

Describe the state of the system at the conclusion of the use case execution. Number each postcondition. Examples:

1. Document contains only valid SGML tags.
2. Price of item in database has been updated with new value.

## 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. This description may be written as an answer to the hypothetical question, “How do I <accomplish the task stated in the use case name>?” This is best done as a numbered list of actions performed by the actor, alternating with responses provided by the system. The normal flow is numbered “X.0”, where “X” is the Use Case ID.

## Alternative Flows

Document other, legitimate usage scenarios that can take place within this use case separately in this section. State the alternative flow, and describe any differences in the sequence of steps that take place. Number each alternative flow in the form “X.Y”, where “X” is the Use Case ID and Y is a sequence number for the alternative flow. For example, “5.3” would indicate the third alternative flow for use case number 5.

## 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. Also, describe how the system is to respond if the use case execution fails for some unanticipated reason. If the use case results in a durable state change in a database or the outside world, state whether the change is rolled back, completed correctly, partially completed with a known state, or left in an undetermined state as a result of the exception. Number each alternative flow in the form “X.Y.E.Z”, where “X” is the Use Case ID, Y indicates the normal (0) or alternative (>0) flow during which this exception could take place, “E” indicates an exception, and “Z” is a sequence number for the exceptions. For example “5.0.E.2” would indicate the second exception for the normal flow for use case number 5.

## 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.

## Priority

Indicate the relative priority of implementing the functionality required to allow this use case to be executed. The priority scheme used must be the same as that used in the software requirements specification.

## Frequency of Use

Estimate the number of times this use case will be performed by the actors per some appropriate unit of time.

## Business Rules

List any business rules that influence this use case.

## 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.

Use Case Template

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case ID: |  | | |
| Use Case Name: |  | | |
| Created By: |  | Last Updated By: |  |
| Date Created: |  | Date Last Updated: |  |

|  |  |
| --- | --- |
| Actors: |  |
| Description: |  |
| Preconditions: |  |
| Postconditions: |  |
| Normal Flow: |  |
| Alternative Flows: |  |
| Exceptions: |  |
| Includes: |  |
| Priority: |  |
| Frequency of Use: |  |
| Business Rules: |  |
| Special Requirements: |  |
| Assumptions: |  |
| Notes and Issues: |  |

# Prototype of the Main Screens go here, where each has a caption describing what is presented.

# Non-Functional Requirements

Describe any constraints or cross-cutting characterstics of the system in a manner that is clear, spectific, and testable. Only present those which are applicable to your system. Categories include but are not limited to:

Security

Reliability

Usability

Cross-Platform Compatibility

Accuracy