Software Requirements and Design Document

For

Group 1

Version 1.0

Authors:
Michael Clark
Phillip Sheng
Jose Ibarra

1. Overview (5 points)

Give a general overview of the system in 1-2 paragraphs (similar to the one in the project proposal).

We will use Flask to launch our web application. We will be using HTML, CSS, and JavaScript for our frontend design and Python for the backend design. SQlite will be used for our database.

2. Functional Requirements (10 points)

*List the **functional requirements** in sentences identified by numbers and for each requirement state if it is of high, medium, or low priority. Each functional requirement is something that the system shall do. Include all the details required such that there can be no misinterpretations of the requirements when read. Be very specific about what the system needs to do (not how, just what). You may provide a brief design rationale for any requirement which you feel requires explanation for how and/or why the requirement was derived.*

1. *Users will be able to place a bet on a sporting event – High*
2. *Users will be able to log in using valid credentials – High*
3. *Users will be able to create an account – High*
4. *Bets and sporting events will be refreshed every 24 hours after a sporting event has concluded – Medium*
5. *Users can access resources for gambling addiction through the homepage – Low*

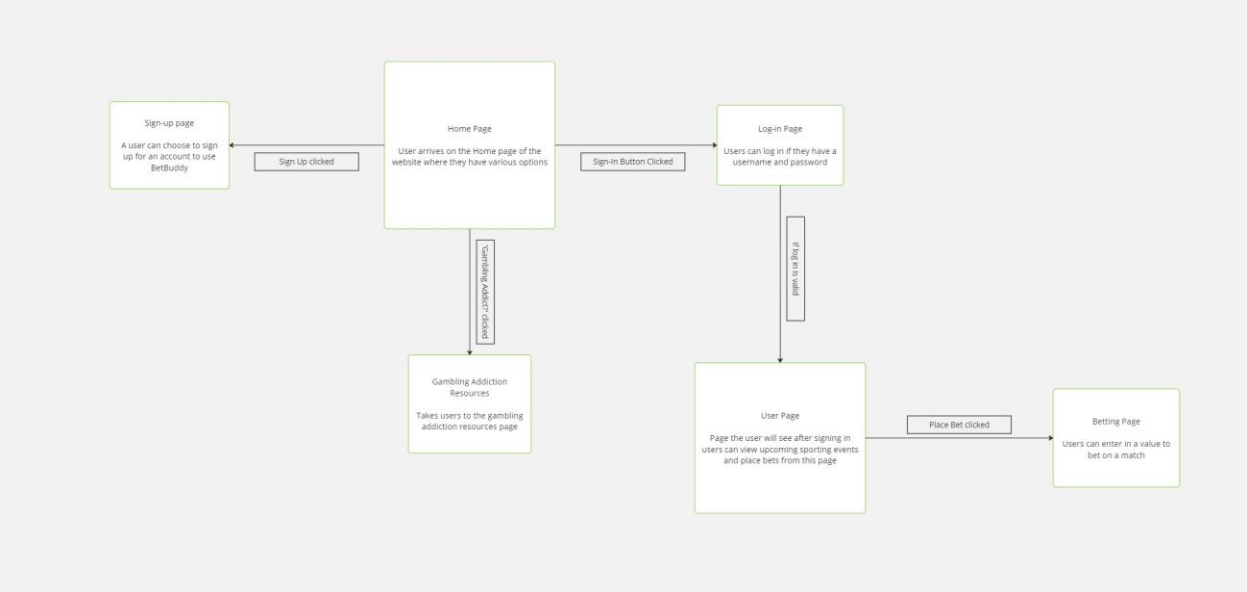
3. Non-functional Requirements (10 points)

*List the **non-functional requirements** of the system (any requirement referring to a property of the system, such as security, safety, software quality, performance, reliability, etc.) You may provide a brief rationale for any requirement which you feel requires explanation as to how and/or why the requirement was derived.*

1. *Usernames will not be repeated.*
2. *Email will not be repeated.*
3. *Ability to find lost password securely*

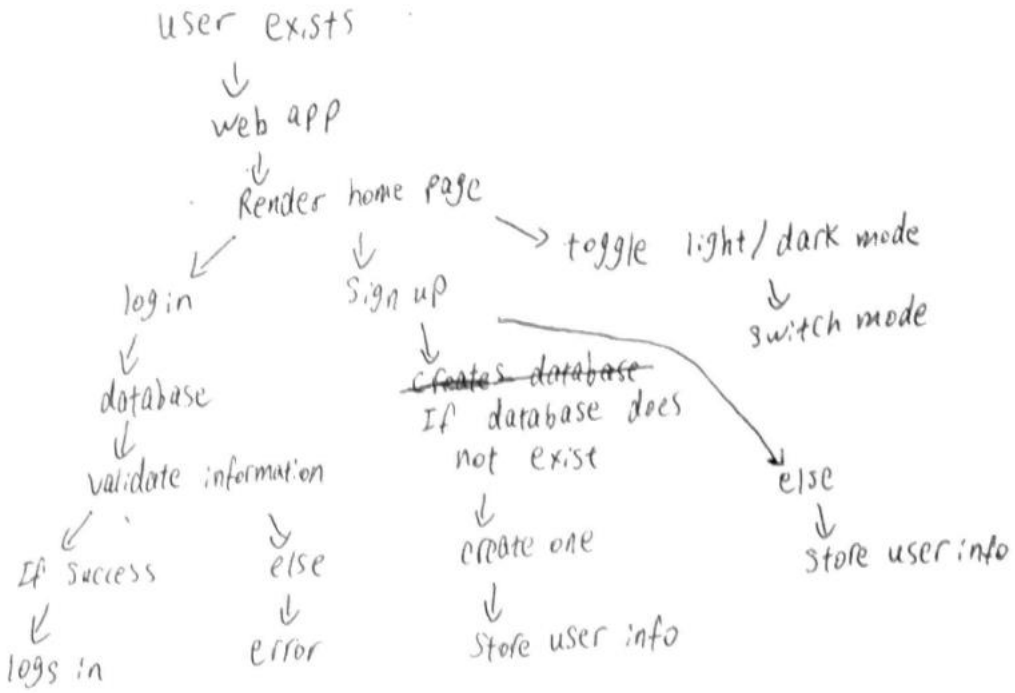
4. Use Case Diagram (10 points)

*This section presents the **use case diagram** and the **textual descriptions** of the use cases for the system under development. The use case diagram should contain all the use cases and relationships between them needed to describe the functionality to be developed. If you discover new use cases between two increments, update the diagram for your future increments.*
***Textual descriptions of use cases**: For the first increment, the textual descriptions for the use cases are not required. However, the textual descriptions for all use cases discovered for your system are required for the second and third iterations.*



5. Class Diagram and/or Sequence Diagrams (15 points)

It is a Sequence Diagram



6. Operating Environment (5 points)

Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.

Windows 10 and 11, MacOS, Linux. Must have Python installed and a working browser. API must also work

7. Assumptions and Dependencies (5 points)

List any assumed factors (as opposed to known facts) that could affect the requirements stated in this document. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project.

API is expired or removed by the third party platform. Flask gets taken off from PyPI, Python gets taken off, a browser update that prohibits user from loading localhost, or an operating system update that affects the use of the computer