Tyler Odom
Nyenty Ayuk-Enow
Amier Chery
Coogan Koerts
Osman Khan

Project Plan Part Two

Schedule:

June 17th - 23rd: Planning and Development of Project Plan

- Milestones include: keeping communication fluid and open throughout these opening phases, submitting all initial documentations on time, establishing a steady workflow

June 24th - 30th: Requirements gathering and analysis

- Milestones include: learning more about requirements gathering and what it means for software, staying organized as the process goes on

July 1st - July 7th: System Design finalization and documentation

- Milestones include: Ensuring continuity and fluidity between the system design and the requirements, ensuring documentation is up to date, preparing for development in a timely manner

July 8th - July 14th: Prototyping and development

- Milestones include: keeping up communication between members, successfully using git for version control, keeping a steady work flow

July 15th - July 21st: Testing and bug-fixes

- Milestones include: Following the test plan, learning more about testing and its importance to the software process

July 22nd - July 26th: Validation and Integration

- Milestones include: preparing for presentation, ensuring that software is portable and usable on different computer/devices with ease, managing dependencies

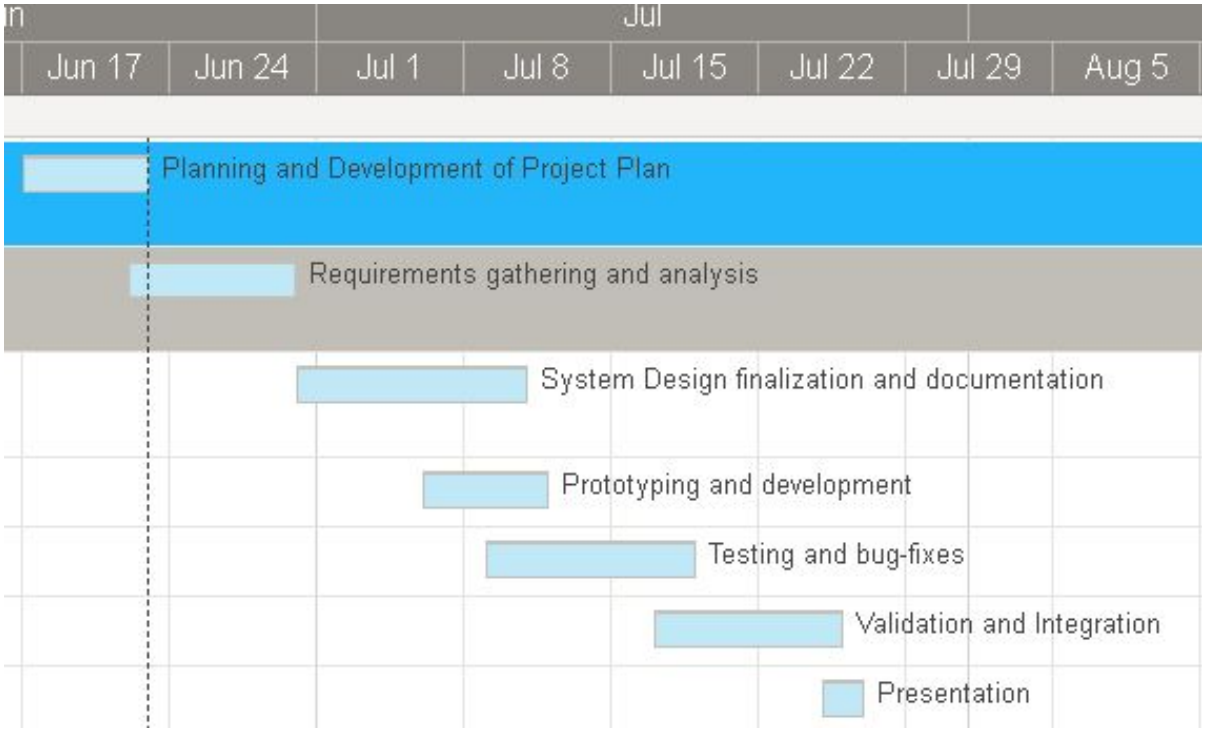

Figure 1. Gantt Diagram of Software Plan

Technical Description:

The Baker's Dozen card game is a dynamic, fluid desktop game developed in Java that simulates the card game Solitaire. This software will utilize the javafx GUI for it's front-end, providing a simple and slick design that maximizes responsiveness in order to give the user a fun and challenging experience.

For now , the game will include only one solitaire game (Baker's Dozen) and will immediately take the user into the game upon opening. The game will include options that give the user the ability to restart, quit out of the game, as well as undo a move. The game will also keep an updated score along the bottom of the screen so the user can track their progress across the duration of the game.

Baker's Dozen card game should be able to run on any desktop device. Due to the simplicity of design , the minimum specifications would be any modern operating system, at least 5 mb of storage, and 2gb of ram. Although this game isn't mobile friendly as of yet, their may be opportunities in the future to develop a more mobile-friendly version.

Test Plan:

The test plan's goal for our software is to ensure that all of the features are working as intended. Multiple tests of each size and scope will be done throughout the development phase to ensure that the software will be developed in a timely manner.

- Features to be tested:
    - Functionality of the game environment (including testing done for different game scenarios)
    - Ensuring that win/lose conditions are properly designed and implemented
    - That cards will work as intended in a stack and recognize when the win conditions are met
    - Additional features such as an Undo button, move counter, and a reset button are properly implemented.
- Approach to Testing
    - A manual approach to testing will be done as it will allow each group member to do independent testing that will allow them to test whatever parts of the code they feel should be tested and come up with solutions on their own. Each group member will be able to be as thorough and detailed as they wish to be.
- Metric for Testing
    - Did we implement the game's rules properly?
        - Does the game recognize a win/lose situation?
        - Does the game stop or allow an illegal move?
    - Are the cards and the values assigned to them properly recognized by the computer?
    - Do our additional features work as intended?
        - Does the move counter count every move?
        - Does the reset button reset the game properly?
        - Does the undo button undo the last move properly?
    - Is the game graphically represented correctly to the user?
- Testing Environment
    - In order to test our code we should use IDE's that are respected and are known to work properly.
        - intelliJ
        - Eclipse
        - NetBeans