**Sudo**

**Project Deliverable 2: Initial Requirements and Modeling**

SOFE2720: Principles of Software and Requirements

Walid Safi 100623815

Hyon Lee 100488156

Diba Shojaei 100621768

Muhammad Ehsan 100634240

Requirements built from the previous deliverable,

1. User starts the game on browser
2. Manager checks the statistics of how many players are playing the game
3. User enters a value in a blank square
4. User selects the difficulty of the game
5. Developer gets feedback from the user
6. User customizes the volume of the game

From the user stories provided from the previous deliverable,

1. As a user of the game, I want to be able to play the game on my web browser
2. As a manager, I want to be able to track how many users are playing the game at a given time
3. As a user of the game, I want the game to inform me when I put incorrect value (unacceptable value) in a blank square
4. As a user of the game, I want to be able to select the difficulty of the game
5. As a developer, I want to receive user feedback, so I can further improve the game
6. As a user, I want to be able to customize the volume of the game

We have narrowed down the importance level as following (In decreasing priority order)

Priority level 1:

* As a user of the game, I want the game to inform me when I put incorrect value (unacceptable value) in a blank square – fundamental of the game
* As a user of the game, I want to be able to play the game on my web browser – User accessibility

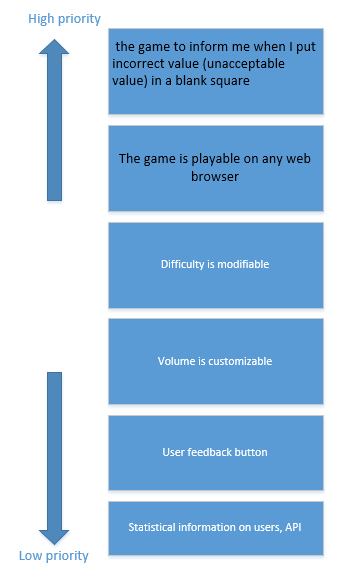
Priority level 2:

* As a user of the game, I want to be able to select the difficulty of the game – User based customization
* As a user, I want to be able to customize the volume of the game

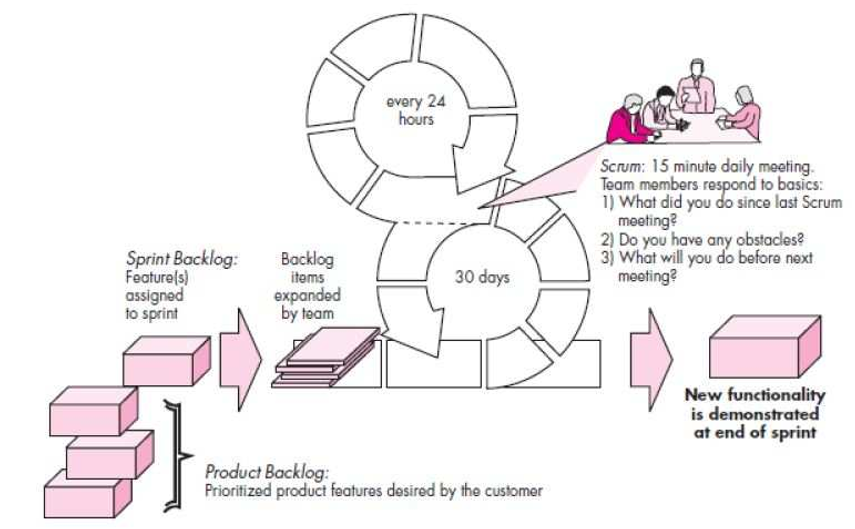
Priority level 3:

* As a developer, I want to receive user feedback, so I can further improve the game
* As a manager, I want to be able to track how many users are playing the game at a given time

The priority formed from the user stories can be represented in a stack (Fig 1).

  
Figure 1. Priority stack to be used for each of the iteration

From the work stack generated, our instinct was to use the scrum process (fig 2)

  
Figure 2. Scrum process

Since the model allows us to work from the high-priority to low (from the core function of the game to accessory functions)

Thus using the following diagram(fig 3.) as the back bone of our iteration cycle,

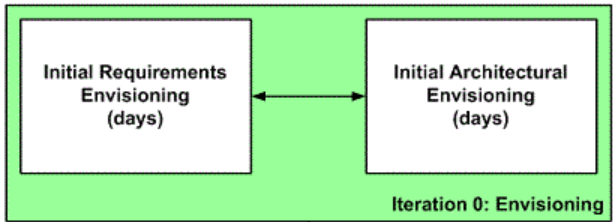
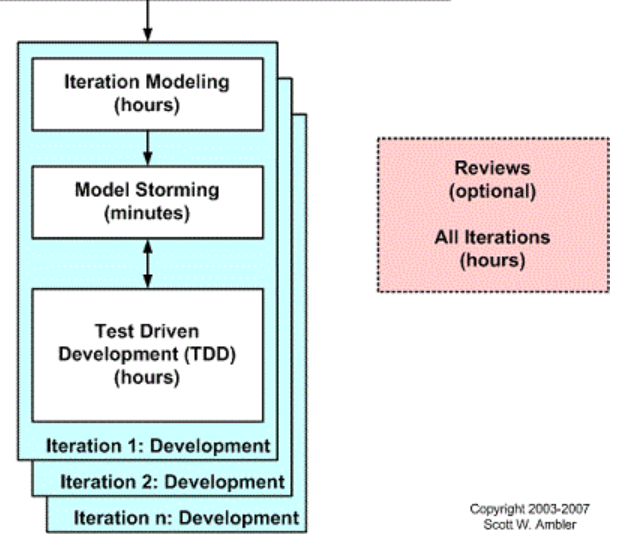
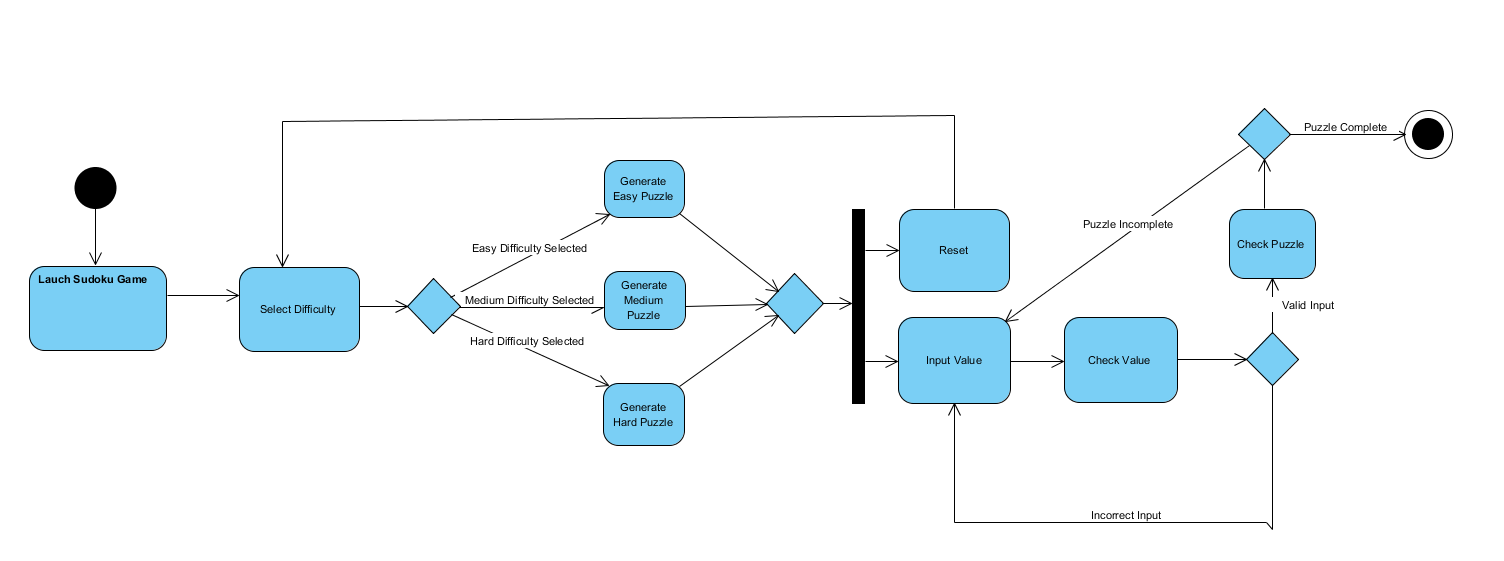
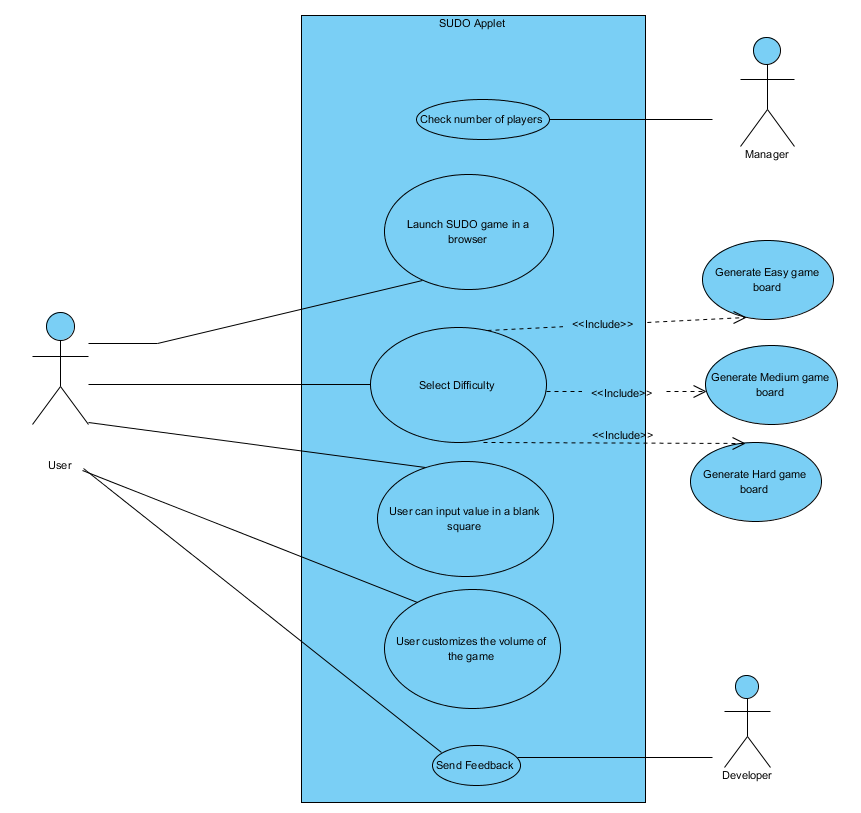


Figure 3. Using the stack generated in figure 1, we will be implementing each iteration as following

**Sequence Diagram**

**Use case Diagram**

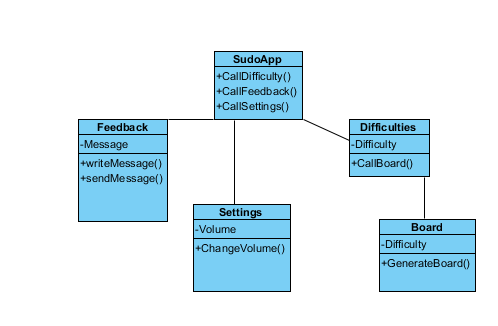


**Conceptual diagram**

SUDO – Sudoku game is an app that is playable in all the browsers. Users can download the game from the app store. Once the game is downloaded, users are guided to the main menu, where they may select the difficulty. The difficulties are separated into easy, medium, and hard mode. Once the difficulty is selected the game generates a 9X9 board with some given numbers. When the user inserts a value of a square, the game would either accept the value if valid, or give out an error if the value is not valid. The user can go into the settings menu, and adjust the volume. The user can also go into the feedback section, and send a feedback to the developers

* Main menu – select different options
* Difficulties – Easy, Medium, Hard
* Settings - change volume
* Board – Checks if the user put a valid input
* Feedback

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Noun Phrases** | **Object** | **Attribute** | **Actors** | **Irrelevant** |
| Main Menu (Sudo App) | Provides connection between all the other classes |  | User | Download |
| Difficulties | Calls the board class to generate the 9X9 board depending on the difficulty | Difficulty (easy, medium, hard) | User |  |
| Settings(Volume) | Controls the volume of game | Volume | User |  |
| Board | Interactive interface for the users | Number of empty squares | User |  |
| Feedback | Communicate between users and developers | Message | Users, developers |  |



**Summary**

We have completed our iteration 0 – the envisioning process. Using the initial requirements and the work stack we have generated from the iteration 0, we will be using sprints on each iteration to add ‘new functionality’ in our game (each functionality can also refer to each entry in our work stack)

Our next step is to move on to our first iteration where we build the fundamental gameplay (our first priority). Unlike other iterations, the first one builds the main function of the game. Therefore we expect the first sprint to take the longest in terms of iteration modeling, model storming, and test driven development.