Software Requirements specification

# Revision history

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Who | Description |
| 1 | 30-06-2012 | Yossi.M. | Initial revision |
| 2 | 30-06-2012 | Yossi.M. | Added section for static architecture  Added section for development path  Added sequence Diagram for SRS-1 |

## Architecture

The following Chart describes the static structure of the system.



* Clients:
  + Android
  + IOS
  + Windows Phone
  + WEB(Html)
* Managers (Services)
  + Session manager
    - Enable the client to perform login operations and User session operations (Debate , Should session be persistent?)
  + Friends manager
    - Enable the client to perform friend related operations.
  + Game Manager
    - Create a game
    - Start a game
    - Make a turn
  + User Info Manager
    - Enable the client to get information about a particular user.
  + Notifications Manager
    - Enable the client to get notifications through a polling mechanism.
* Engines (Services or Components):
  + Persistent Game State engine – (Debate : Service or DLL?)
    - Enable the user to write and restore a Game state.
  + Persistent user information engine (Debate , Service or DLL?)
    - Enable a use to write and restore user information
* Resources (Core) (Services):
  + -Facebook Access – An internal representation for the Facebook API
  + -Storage – An internal representation for storage
* Utilities
  + Log Book – Logging Service
  + Publisher/Subsriber – Will by the notification manager.

# Development Path



# Software Requirements Specifications

|  |  |
| --- | --- |
| **SRS-1** | |
| **Description** | The user shall be able to login to the System using his Facebook credentials. |
| **Traced from** | URS 3.1 |
| **User Story** | 1. The user activate the client 2. The Client shows to the user a log in screen 3. Evaluate credentials :  If the user doesn’t have Login credentials stored locally at his machine/device :    * Client Show the user a login screen.    * The user enters his credentials.    * The user click on a button to proceed.    * The client Store the login credentials locally on the client machine. 4. The client process the request and login the user to the system, using the credentials. 5. The Client receives a session token. |
| **UI** | TBD |
| **System Sequence Diagram** |  |
| **Other Design Comments** |  |

|  |  |
| --- | --- |
| **SRS-2** | |
| **Description** | The user shall have a list of friends. The user shall be able to observe his list of friends at any time. |
| **Traced from** | URS-4.1 , URS 4.2 |
| **User Story** | Assumption   * The user is logged in to the system and the client is at the foreground of his screen.   User Story   1. The user click on button that take him to a friend’s screen. 2. The user sees a list of his friends. |
| **UI** | TBD |
| **System Sequence Diagram** |  |
| **Other Design Comments** |  |

|  |  |
| --- | --- |
| **SRS-3** | |
| **Description** | A user shall be able to import a subset of Friends from his Facebook account friends. |
| **Traced from** | URS 4.3 |
| **User Story** | Assumption   * The user is logged in to the system and the client is at the foreground of his screen. The friend screen is showed to the user.   User Story   1. The user Click on a button that take him to the friends import screen 2. The client show to the user a list of his Facebook friends 3. The user choose which users he would like to import and click a button to proceed 4. The friends import window is closed. 5. The user sees the friends Screen with the newly imported friends. |
| **UI** | TBD |
| **System Sequence Diagram** |  |
| **Other Design Comments** |  |

|  |  |
| --- | --- |
| **SRS-4** | |
| **Description** | Each User shall be able to create a new game |
| **Traced from** | URS 5.1, URS-Game-Description |
| **User Story** | Assumption   * The user is logged in to the system and the client is at the foreground of his screen.   User Story   1. The user click on a button that take him to the Game creation screen 2. At the Game creation screen , the user set:    1. A Title for the game (suggest a default of something that auto generated)    2. Participants    3. Number of cycles (suggest a default of 1)    4. Minimum number of characters per turn (It shall suggest a default) 3. The user click on the create button 4. The game is registered to the system. |
| **UI** | TBD |
| **System Sequence Diagram** |  |
| **Other Design Comments** |  |

|  |  |
| --- | --- |
| **SRS-4** | |
| **Description** | Each User shall be able to start a new game |
| **Traced from** | URS 5.1, URS-Game-Description |
| **User Story** | Assumption   * The user is logged in to the system and the client is at the foreground of his screen. The client is showing a game screen.   Story   1. The user click on a start button 2. The client Ask the user to fill the opening phrase. 3. The user fill the phrase 4. The client of the next user (this is another client instance hosted on another client machine) pop a notification to the user that this is his turn. |
| **UI** | TBD |
| **System Sequence Diagram** |  |
| **Other Design Comments** |  |

|  |  |
| --- | --- |
| **SRS-5** | |
| **Description** | Each User shall be able to make a turn in a game in which he participate |
| **Traced from** | URS-Game-Description |
| **User Story** | Assumption   * The user is logged in to the system and the client is at the foreground of his screen. The user got a notification telling him this is turn.   Story   1. The user opens the notification dialog. 2. The user Click on a button allowing him to make his turn 3. The client show to the user the opening words of the previous phrase and ask the user to fill a phrase. 4. The user fills a phrase and fills opening words for the next phrase. 5. If the turn is the last in the game, the client won’t ask for the opening words of the next words. 6. The user click on a button to end the turn. 7. If the turn is the last in the game, All Participant get a notification that the game was ended. 8. If the turn in not the last in the game, the next participant get a notification about his turn. |
| **UI** | TBD |
| **System Sequence Diagram** |  |
| **Other Design Comments** |  |

|  |  |
| --- | --- |
| **SRS-6** | |
| **Description** | The client shall allow users to observe old stories in which they have participated. |
| **Traced from** | URS-5.3 |
| **User Story** | Assumption   * The user is logged in to the system and the client is at the foreground of his screen.   Story   1. The User click on a button taking him to the Games History screen 2. The client show to user a list of games he was participating in. The list is ordered in chronological ordering, Most Recent story first. 3. The user click on a specific game ( a Specific line in the list) 4. Another screen is opened, showing to the user the story and the participants of the game. |
| **UI** | TBD |
| **System Sequence Diagram** |  |
| **Other Design Comments** |  |