Interface Specification

<Module Name>

# Introduction

<Specify the aim of the document from the module’s point of view.>

# Services

<Specify the services provided and/or required by the module.>

## Services Provided

|  |  |  |
| --- | --- | --- |
| Service | Provided By | Tested By |
| User can write an email to other players | writeText, selectRecipients | T1 |
| User can check inbox to read the emails for him/her. | readEmail | T2 |

## Access Method

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Access Method** | **Parameter name** | **Parameter type** | **Description** | **Exceptions** | **Map to services** |
| writeText | text | String | text is String type |  | 1 |
| selectRecipients | recilist | array of String | recilist is an array of String which indicates the recipients |  | 1 |
| readEmail | emaillsit | array of Email | emaillist is an array of Email which shows a list of emails. |  | 2 |
| sendEmail | anEmail | Email | integrate the information in Method writeText and selectRecipients to form an Email, and send out the Email to DB of emails. |  | 1 |

## Access Method Effects

|  |  |
| --- | --- |
| **Access Method** | **Description** |
| writeText | User can write content in the textbox in order to send the email. Store the text as *String* type and then the text will be used when sending emails to others. |
| selectRecipients | We use an *array of String* recilist to store the recipients list. User can select recipients and once a recipient is chosen, we add this recipient into the array of String. |
| readEmail | Check the database of email and filter out all the emails whose recipients list has the user’s name. Then return an *array of Email* type emaillist. |
| sendEmail | Receive the parameter text and recilist to form anEmail, which is a type of *Email.* Then send it to DB of emails. |

## Services Required

<Format similar to Sec. 2.1.>

*none*

# Local Types

<Specify the data formats inside/between the module(s).>

|  |  |
| --- | --- |
| **Type** | **Value Space** |
| Identification |  |
| Command |  |

# Interface Design Issues

<Describe any design issues that arose during development. Describe the alternatives and the rationale for the alternative chosen.>

Eg. 1. Whether there is a database of email we can store all the emails there?

# Test Cases

<Characterize the expected value of the outputs over sets calls to the module.>

### T1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | writeText | String | Email content is stored in parameter text and ready for the future use. |  | 1 |
| 2 | selectRecipients | array of String | Recipients list is made successfully in parameter recilist and ready for the future use. |  | 2 |
| 3 | sendEmail | String text and array of String recilist | Construct anEmail by using input text and recilist and sent anEmail to the DB of emails. |  | 3 |

### T2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step** | **Description** | **Input Type/Value** | **Expected Results** | **Service** | **Preamble** |
| 1 | readEmails |  | An array of Emails are returned. |  | 1 |