Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <22/11/2014> | <1.0> | Added Introduction and Software Architecture Overview | Bahadır Kırdan, Ömer Emre Aslan, Burak Atalay |
| <23/11/2014> | <1.1> | Added sequence diagrams | Yunus Yılmaz, Vidal Hara |
| <24/11/2014> | <1.2> | Components Updated | Burak Atalay, Ömer Emre Aslan |
| <01/12/2014> | <1.3> | Components Updated | Burak Atalay, Ömer Emre Aslan |
| <01/12/2014> | <2.0> | Final Edits | Burak Atalay |

**TABLE OF CONTENTS**

Revision History 1

1 Introduction 3

1.1 References 3

1.1.1 Project References 3

2 Software Architecture overview 3

3 Software design description 3

3.1 Graphical User Interface 4

3.1.1 Component interfaces 4

3.1.2 Component design description 5

3.1.3 Workflows and algorithms 6

3.1.4 Software requirements mapping 8

3.2 Controllers 8

3.2.1 Component interfaces 8

3.2.2 Component design description 8

3.2.3 Workflows and algorithms 9

3.2.4 Software requirements mapping 10

3.3 Model 10

3.3.1 Component Interfaces 10

3.3.2 Component Design Description 11

3.3.3 Software Requirements Mapping 11

# Introduction

This document describes the design of the OzuSch software system.

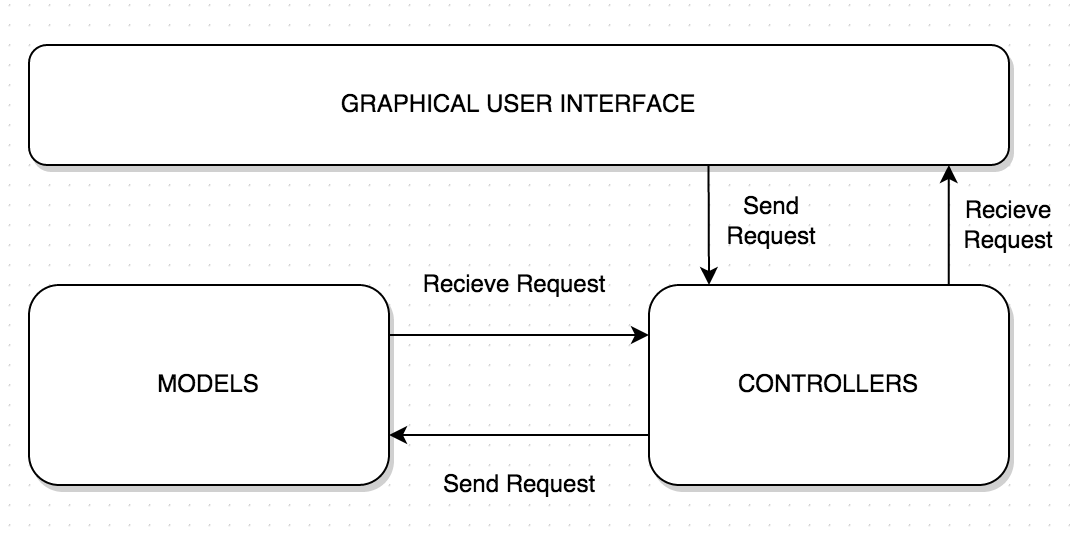
## References

### Project References

| # | Document Identifier | Document Title |
| --- | --- | --- |
| 1 | Doc #1 | Software Development Plan |
| 2 | Doc #2 | Software Requirements Specifications |

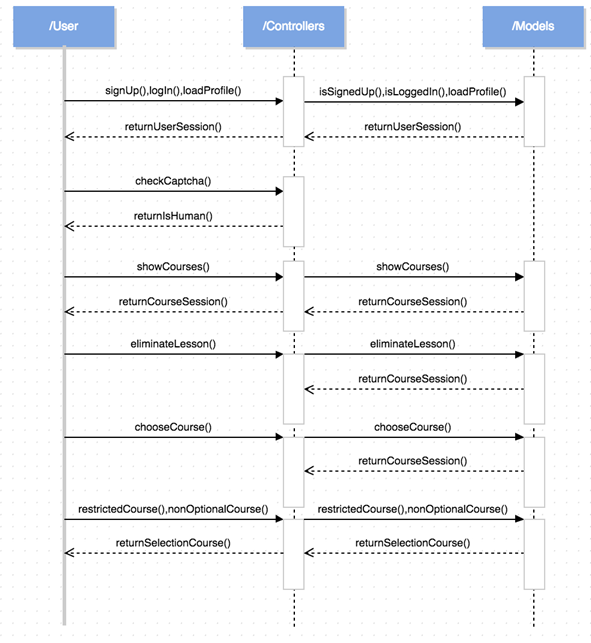
# Software Architecture overview

In this project -OzuSch- we implement system using Model-View-Controller architectural design pattern. Graphical User Interface part include only View component. Front End Services get the data from GUI and send these data to Controller. Controller sends data request to Model.



# Software design description

* **Graphical User Interface:** It is responsible for displaying a given formatted data. Formatted data can be a string object. Also it directs user actions to Controller component.
* **Controller:** It analyzes the data retrieved from GUI component. The controller interacts with the model that it is related with. The received data from model are sending to GUI component.
* **Model:** It is the component that all data is stored and handled.



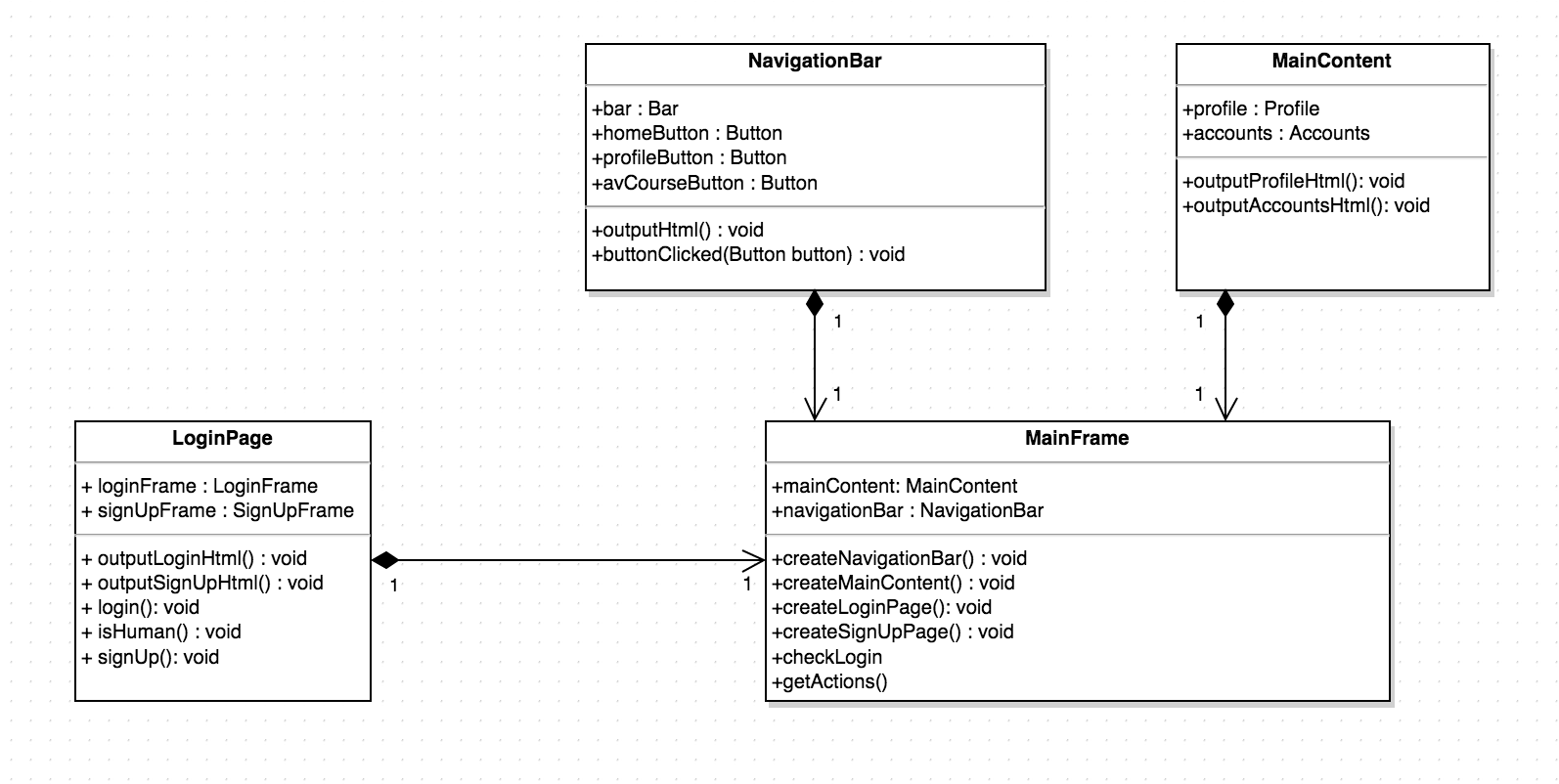
## Graphical User Interface

### Component interfaces

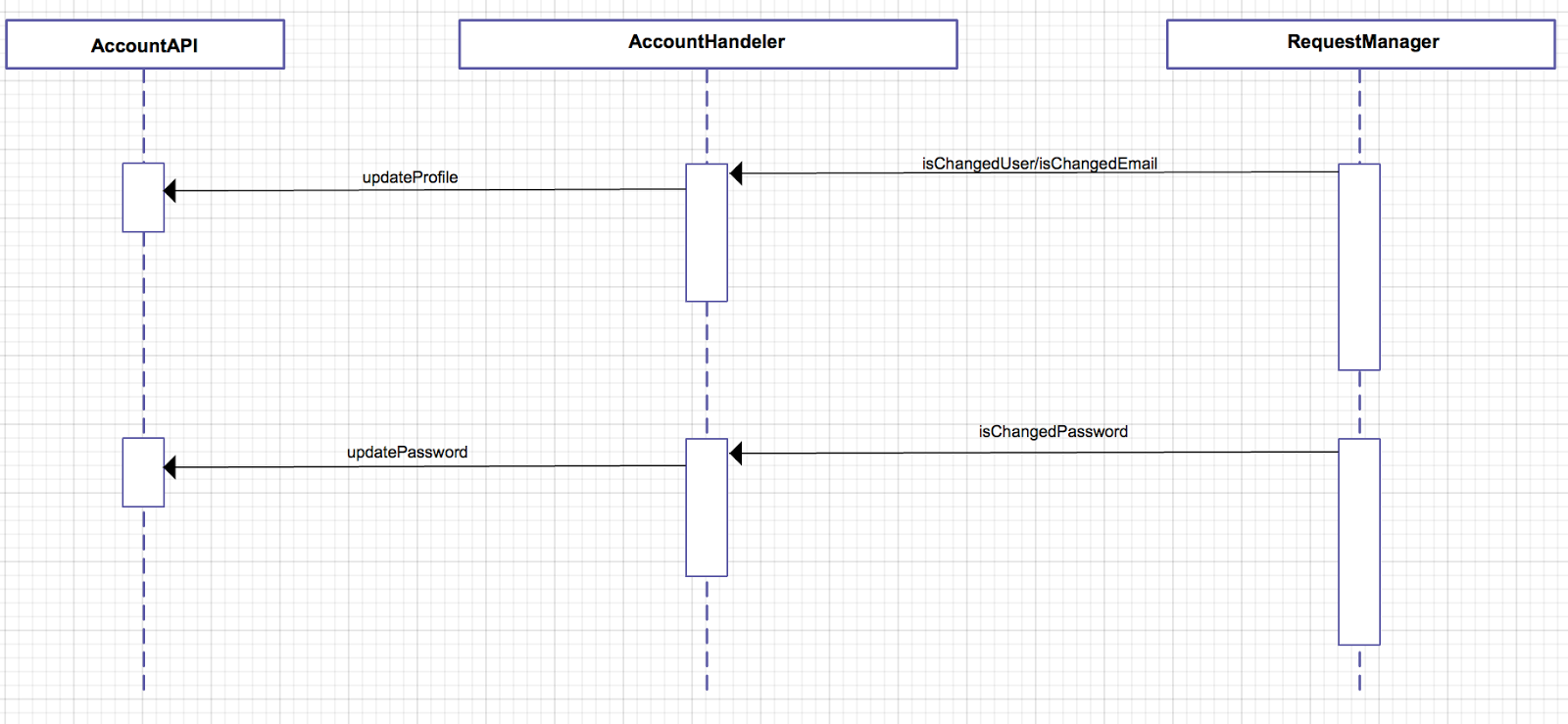
MainFrame is the base class of the Graphical User Interface. It includes three main components, which are NavigationBar, MainContent and LoginPage

* **NavigationBar**
  + **NavigationBar** is located at the top of the MainFrame and it contains these components:
    - Home Button
    - Profile Button
    - Available Courses Button
  + **MainContent** is an interface for everything that can be shown in MainFrame. These contents are possible content types:
    - Profile
    - Accounts
  + **LoginPage** is an interface for the user to enter his/her account.
    - SignUp Button
    - SignIn Button

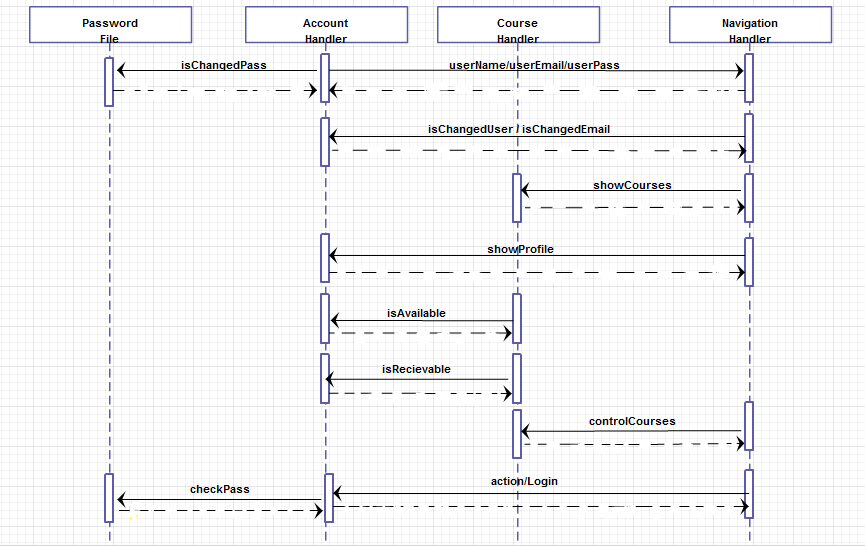
### Component design description

****

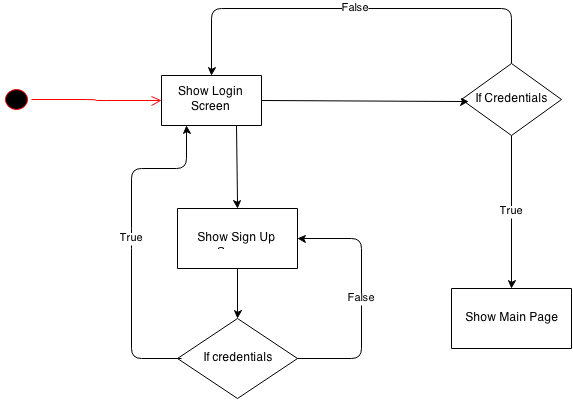
### Workflows and algorithms



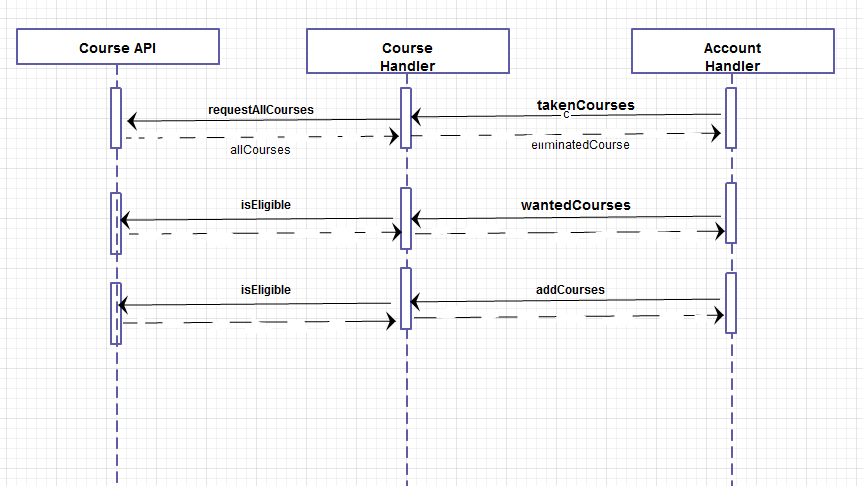
Sequence Diagram between Controller and Model



Sequence Diagram between Model and Controller



Start Diagram for View Component



Sequence Diagram between Model and Controller

### Software requirements mapping

* SRS-OZUSCH-001
* SRS-OZUSCH-004
* SRS-OZUSCH-006

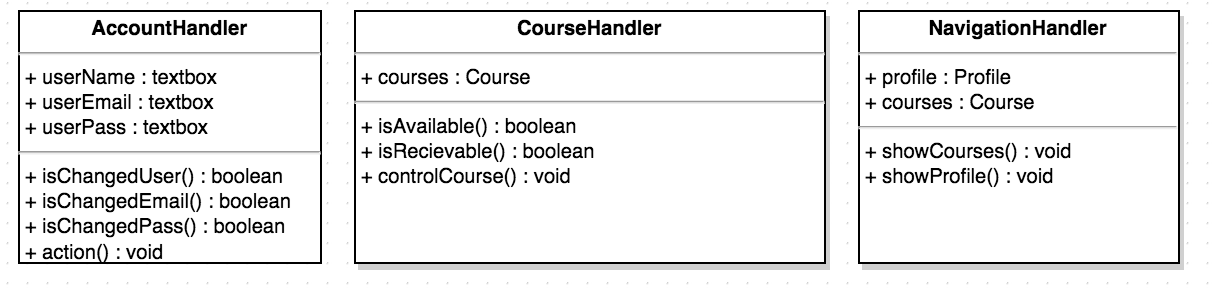
## Controllers

It sends the data retrieved from Graphical User Interface to Model component for analyzing and processing.

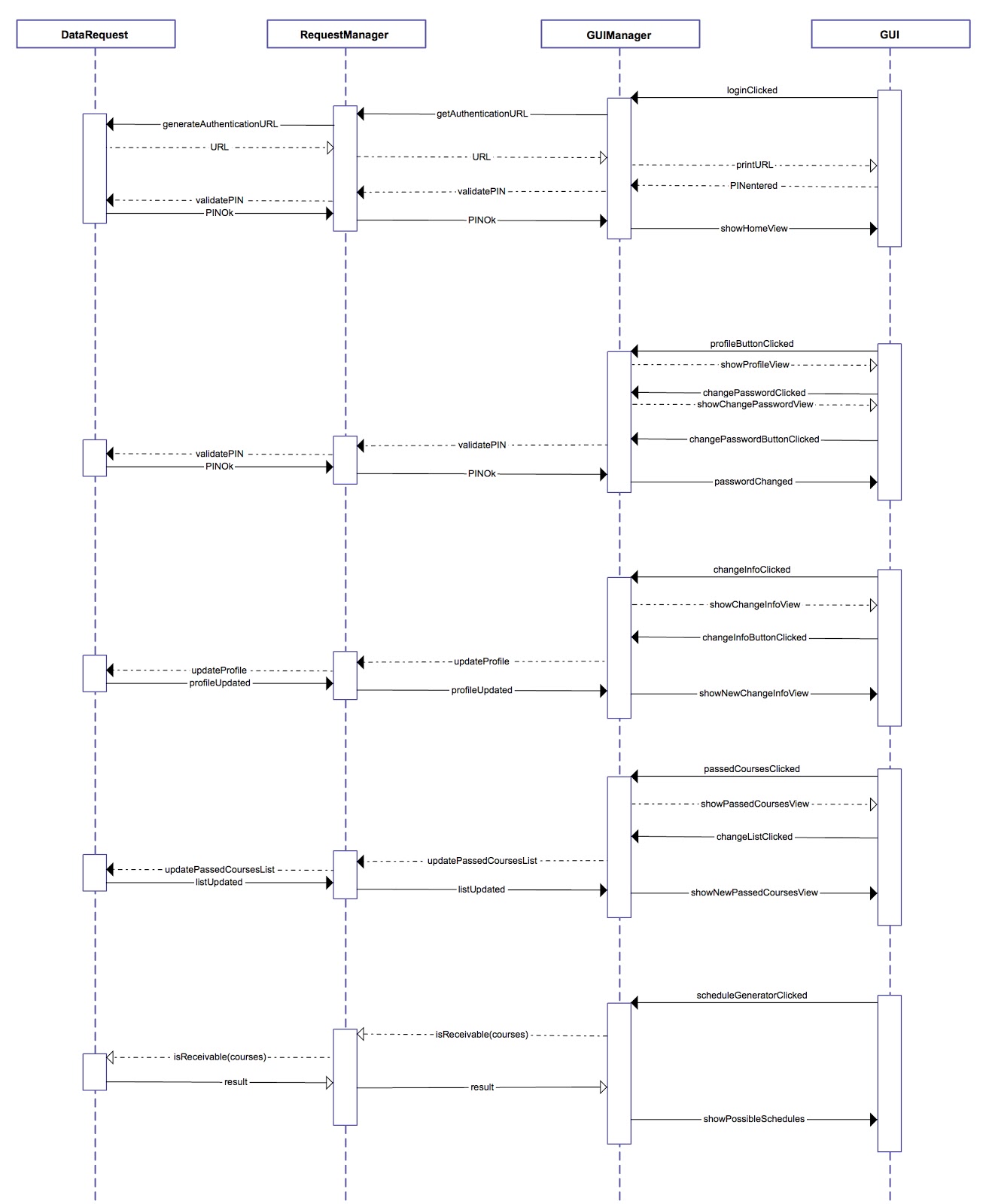
### Component interfaces

* AccountHandler
  + Responsible for changing the user name, user email and courses that are taken by user.
* NavigationHandler
  + Responsible for navigating throughout views
* CourseHandler
  + Responsible for controlling courses that are available and receivable

### Component design description

****

### Workflows and algorithms



General Sequence Diagram between 3 Major Components

### Software requirements mapping

* SRS-OZUSCH-002
* SRS-OZUSCH-003
* SRS-OZUSCH-007
* SRS-OZUSCH-011
* SRS-OZUSCH-012

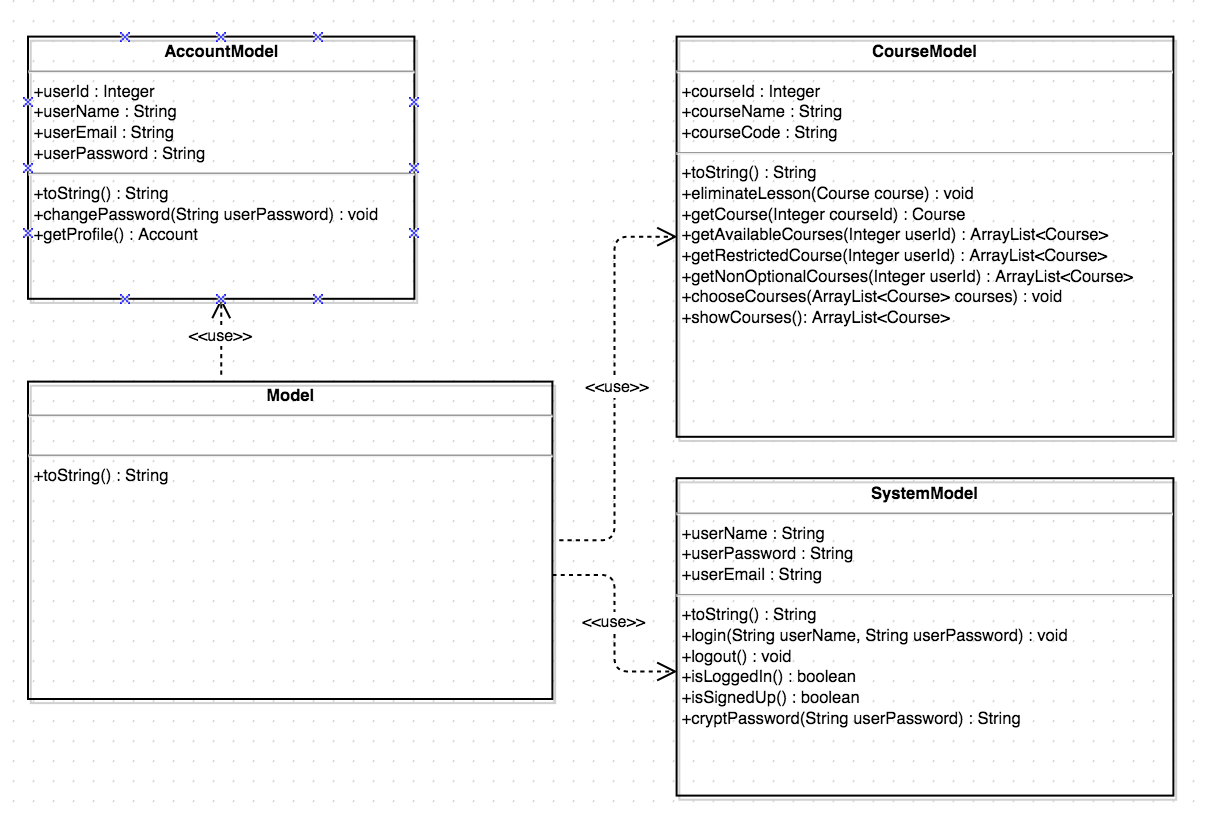
## Model

It is the component that all data is stored and handled.

### Component Interfaces

* AccountModel
  + Responsible for changing the user name, user email and courses that are taken by the handler.
  + It stores and returns profile as Account object
* CourseModel
  + Responsible for handling course processes
  + It stores and returns profile like restricted courses, available courses etc.
  + Eliminate course method provides handling
  + Adding courses to basket and selecting them
* SystemModel
  + Responsible for main system processes like login/logout
  + Crypt the password for security

### Component Design Description

****

### Software Requirements Mapping

* SRS-OZUSCH-005
* SRS-OZUSCH-008
* SRS-OZUSCH-009
* SRS-OZUSCH-010
* SRS-OZUSCH-013
* SRS-OZUSCH-014
* SRS-OZUSCH-015