Software Design Specification

(SDS)

Rabt Website Application

Version 1.0

Prepared by Fizza Shafiq, Umer Ahmad Cheema & Ismail Tirmizi

NUST-SEECS

1 Jab 2017

Contents

[1. Introduction 4](#_Toc469909916)

[1.1 System Purpose 4](#_Toc469909917)

[1.2 System Scope 4](#_Toc469909918)

[1.3 Definitions, Acronyms, and Abbreviations 5](#_Toc469909919)

[2. System Overview 5](#_Toc469909920)

[3. Analysis Model 6](#_Toc469909921)

[3.1 Interface Objects 6](#_Toc469909925)

[3.1.2 Acquaintance Association interface objects 7](#_Toc469909926)

[3.1.3 Containership 7](#_Toc469909927)

[3.1.3.2 Add Interface 8](#_Toc469909928)

[3.1.3.3 Update Interface 8](#_Toc469909929)

[3.1.3.4 Delete Interface 9](#_Toc469909930)

[9](#_Toc469909931)

[3.2 Sub System Diagram 11](#_Toc469909932)

[3.3 Class Diagram 12](#_Toc469909933)

[3.4 Control Objects 13](#_Toc469909934)

[4. Design Model 16](#_Toc469909936)

[4.1 Sequence Diagrams 16](#_Toc469909937)

[4.1.1 Database Population 16](#_Toc469909940)

[4.3.2 Policy Creation 21](#_Toc469909944)

[4.3.3 Policy Generation 27](#_Toc469909945)

[4.3.4 Manage Pre-Updates 28](#_Toc469909947)

[4.3.5 Manage OnGoing-Updates 28](#_Toc469909948)

[4.3.6 Manage Post-Updates 30](#_Toc469909949)

[4.3.7 Request Interception 31](#_Toc469909950)

[4.3.8 Attribute Update 32](#_Toc469909951)

[4.3.9 Access Request Re-evaluation 33](#_Toc469909952)

[4.3.10 Policy Evaluation 34](#_Toc469909953)

[4.4 Communication Diagrams 35](#_Toc469909954)

[4.4.1 Database Population 35](#_Toc469909955)

[4.4.2 Policy Creation 38](#_Toc469909956)

[4.4.3 Policy Generation 42](#_Toc469909957)

[4.4.4 Manage Pre-Updates 43](#_Toc469909958)

[4.4.5 Manage On-Updates 43](#_Toc469909959)

[4.4.6 Manage Post Updates 44](#_Toc469909960)

[4.4.7 Request Interception and Policy enforcement 44](#_Toc469909961)

[4.4.8 Enforce Attribute Update 45](#_Toc469909962)

[4.4.9 Access Request Re-Evaluation 46](#_Toc469909963)

[4.4.10 Policy Evaluation 47](#_Toc469909964)

[4.5 Activity Diagrams 48](#_Toc469909965)

[4.6 Deployment Diagram 48](#_Toc469909966)

[4.7 Component Diagram 48](#_Toc469909967)

[4.8 Package Diagram 48](#_Toc469909968)

# 

# Introduction

## System Purpose

After completing the SRS document we are done with the requirement phase of the project so next we must move onto the architecture and design phase of the document. This SDS document will serve as a means to explain minute details of the architecture and design of our ‘Rabt’ Website Application. To present our work we have decomposed the project into numerous components and this document will dive into detail to explain them. The top level architecture of the system gives a higher level overview, design strategies, detailed system design, various design views, UML diagrams and deployment architecture that will be described in this document.

## System Scope

The ‘*Rabt’* Website Application will extend its use to those users who are excessively involved in project development. They will be able to find experts and people with experience in the project field that they are currently working on. These experts can then help the project developers in many ways. The web application is developed to assist the students in universities and the freelancer community, which are mostly in need of guidance and abstract help in the start of the project. When these guys dive in their questions and their problems become very much bespoke and specific they require people who have previously done such projects for help. Our web app plays a critical role in finding such experts for the users. The website application also provides the user an interface where when a user is done with his project, they can also promote his status as an expert in that project ensuring the return of the help he received from the site.

The application will assist user by giving them accounts and working space where they can organize and optimize their work. It also includes module which will allow users to set deadlines for their work, upload project files to share and a newsfeed area to communicate with multiple people.

This *Software Design Specification* document describes the details of the system design decisions. The design of graphical user interface for the system administration and CSCs is discussed in Software Requirement Specification document, so they are not addressed in this document.

## Definitions, Acronyms, and Abbreviations

|  |  |
| --- | --- |
| **API** | Application Programming Interface |
| **JDBC** | Java Database Connectivity |
| **SDS** | Software Design Specification |
| **SQL** | Structured Query Language |
| **SRS** | Software Requirements Specification |
|  |  |

# System Overview

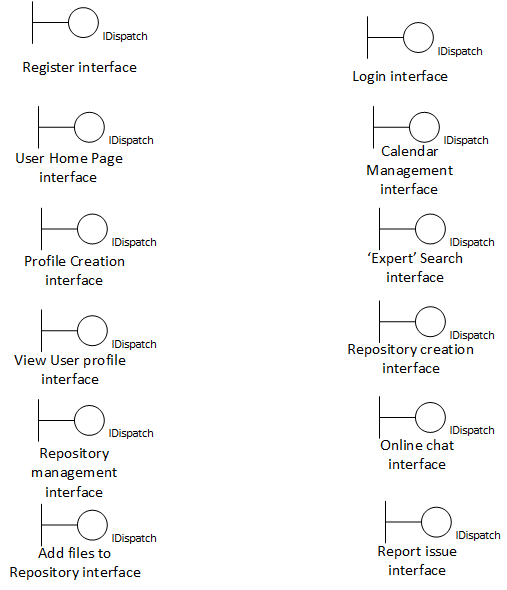
‘Rabt’ website will be launched as a Web Application that will work on a single server with clients scattered around. The client will only require a working web browser on the working device. The application will make use of JDBC driver for the connection to the MySQL database. For access to the resources the client will require an internet connection. The ‘Rabt’ server will be required to be connected to the internet for the transactions.

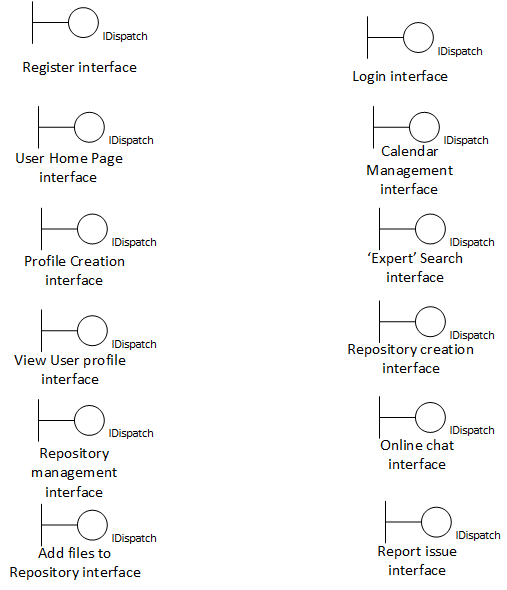
# Analysis Model



## Interface Objects

The interface objects represent the main interfaces of the system. Given below is the list of different interface objects used in our system.





### User cases Interface objects

#### Add Repository



Delete Repository



Register and Login



## Class Diagram



## Control Objects

Control objects are responsible for computation or processing tasks. Following are the main control objects in the proposed framework.



**Description of Control Objects**

1. **Navigation Controller:**

This control object is responsible for the navigation among interfaces.

1. **Subject Controller:**

This control object is used to manage policy *Subject* parameter. It handles all the Subject Add, Update and Delete operations as per user request.

1. **Action Controller:**

This control object is used to manage policy *Action* parameter. It handles all the Action Add, Update and Delete operations as per user request.

1. **Resource Controller:**

This control object is used to manage policy *Resource* parameter. It handles all the Resource Add, Update and Delete operations as per user request.

1. **Environment Controller:**

This control object is used to manage policy *Environment* parameter. It handles all the Environment Add, Update and Delete operations as per user request.

1. **Condition Controller:**

This control object is used to manage policy *Condition* parameter. It handles all the Condition Add, Update and Delete operations as per user request.

1. **Target Controller:**

This control object is used to manage policy *Target* parameter. It handles all the Target Add, Update and Delete operations as per user request.

1. **Rule Controller:**

This control object is used to manage policy *Rule* parameter. It handles all the Rule Add, Update and Delete operations as per user request.

1. **Obligation Controller:**

This control object is used to manage policy *Obligation* parameter. It handles all the Obligation Add, Update and Delete operations as per user request.

1. **Policy Controller:**

This control object is used to manage policy *Policy* parameter. It handles all the Policy Add, Update and Delete operations as per user request.

1. **Policy Set Controller:**

This control object is used to manage policy *Policy Set* parameter. It handles all the Subject Add, Update and Delete operations as per user request.

1. **Response Receiver:**

This control object is used to receive XACML based policy response from PDPaaS component.

1. **Response Sender:**

This control object is used to send XACML based policy response to PEPaaS component.

1. **Request Receiver:**

This control object is used to receive XACML based policy request at PDPaaS component.

1. **Request Sender:**

This control object is used to send XACML policy request from PEPaaS component.

1. **Policy Sender:**

This control object is used to send XACML policy from PAPaaS to policy repository.

1. **Policy Receiver:**

This control object is used to receive XACML policy from PAPaaS in policy repository.

1. **Attribute Update Controller:**

This control object is used to update the values of attributes in UCON specific Pre, Post and OnGoing Update requests.

1. **Request Evaluator:**

This control object is used to evaluate the XACML policy evaluation request.

1. **Usage Monitor:**

This control object is used to invoke the access request re-evaluation request in case of OnGoing and Post attribute updates.

1. **Policy Generation Controller:**

This control object is responsible for the generation and storage of XACML based policy.

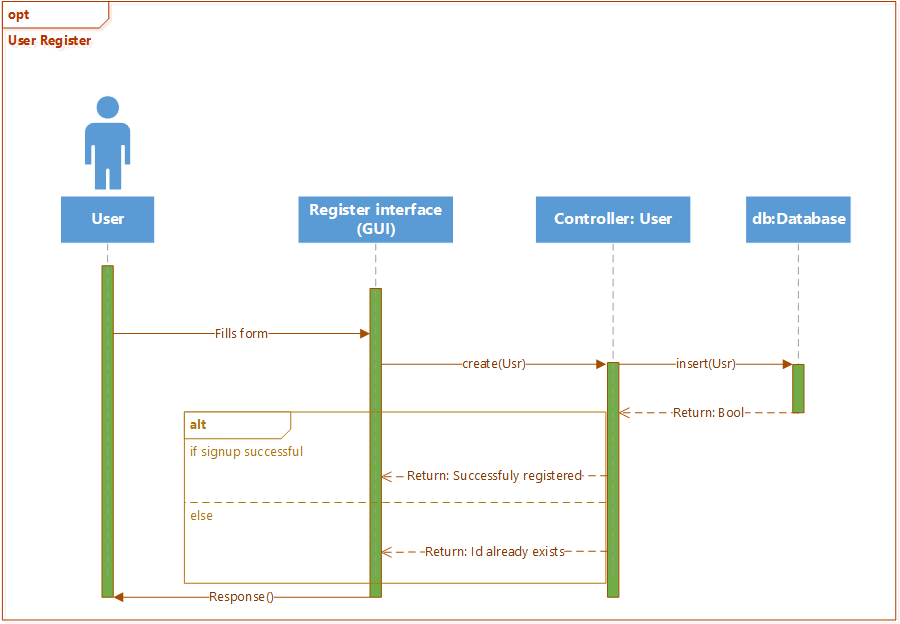


# Design Model

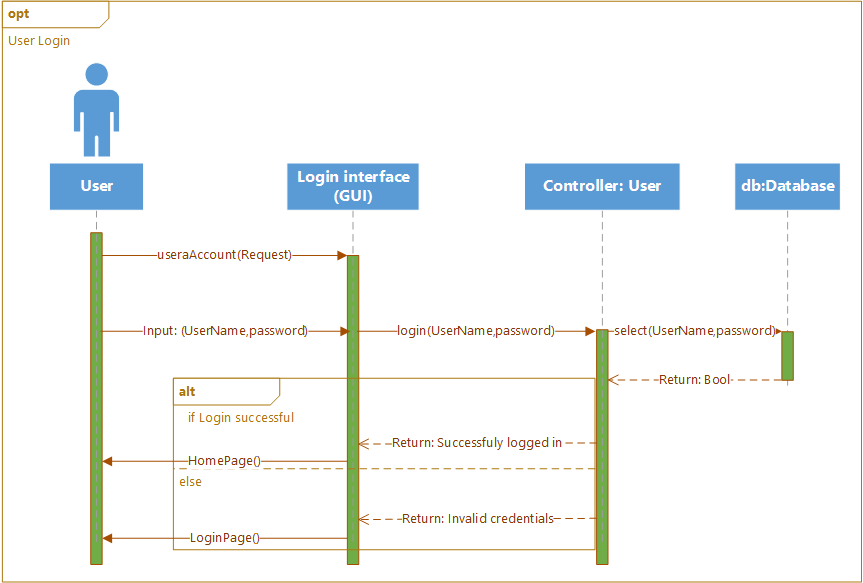
## Sequence Diagrams



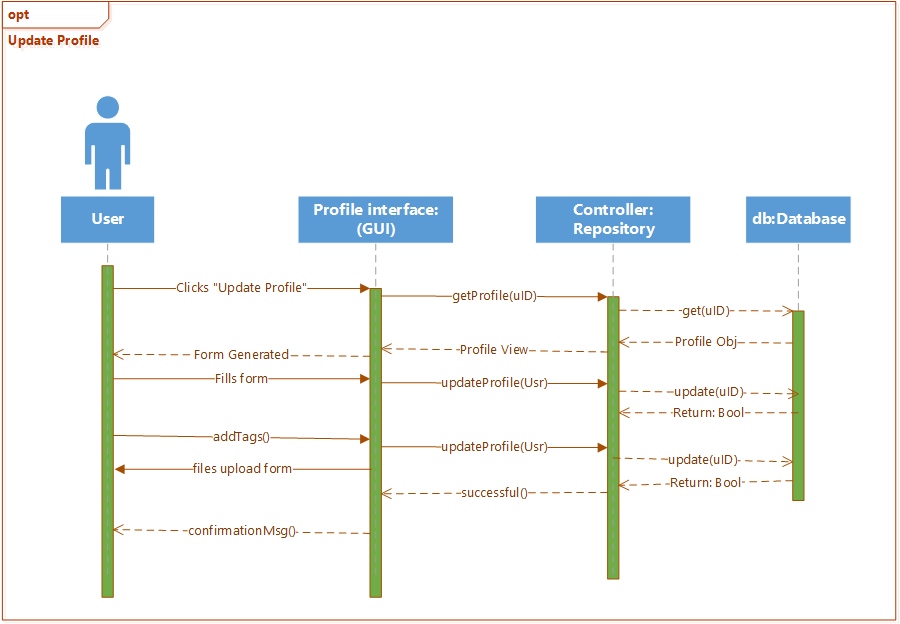
### Registration



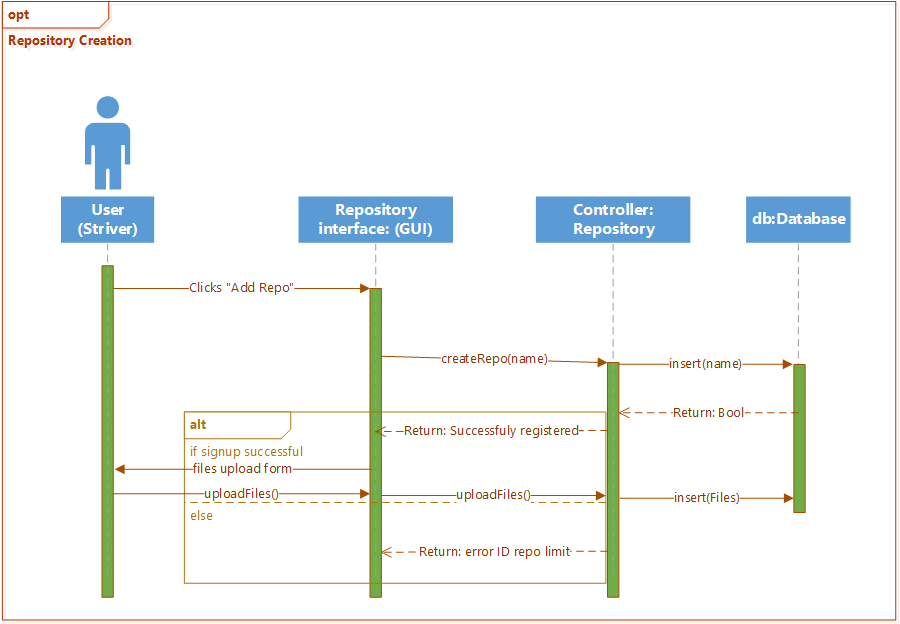
### Login



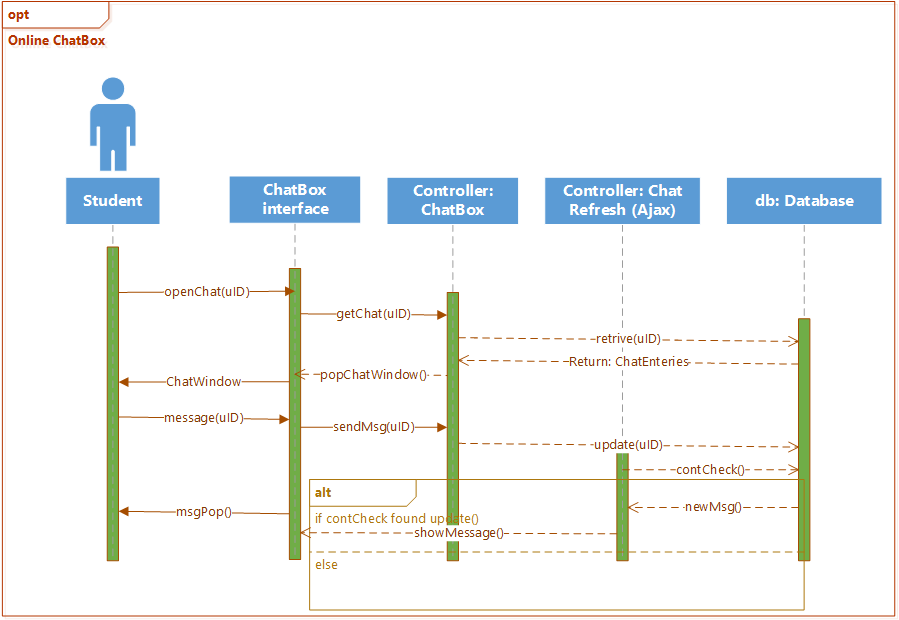
### Profile Creation



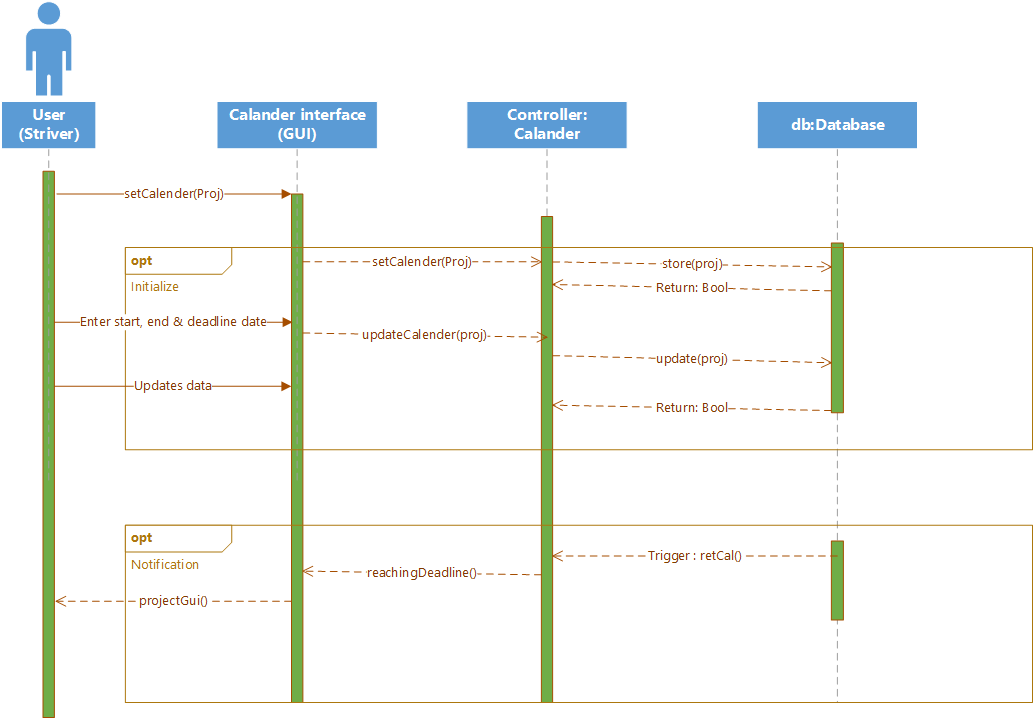
### Repository Creation



### Online Chat



### Calendar Management



## Communication Diagram

### Registration



### Login



### Profile Creation



### Repository Creation



### Calendar Management



### Chat Box





## Activity Diagrams



## Deployment Diagram



## Component Diagram



## Package Diagram

