A5 DD-1
STUDENT INFO

Serik & Co: Team-2

Design Document for Student-Info project

Group#2

Serik: %marks

Alikhan: %marks

Arman: %marks

Mereke: % marks

Akbota: % marks

Version	Date	Author	Change
0.1	20/03/15		Initial Document

Table of Contents

1 Introduction

1.1 PURPOSE

The purpose of this document is to explain the design and achitecture of the Student-Info application

1.2 SCOPE

This document covers system decomposition, interfaces, and dependencies, as well as design rationale

1.3 DEFINITIONS, ACRONYMS, ABBREVIATIONS

Term	Description
Login page	First page of our web application. When user enters address of our web application user gets this page
Main page	Page which user gets after entering correct data to login page
Filter	Used to simplify search queries by checking some checkboxes or radio buttons

1.4 DESIGN GOALS

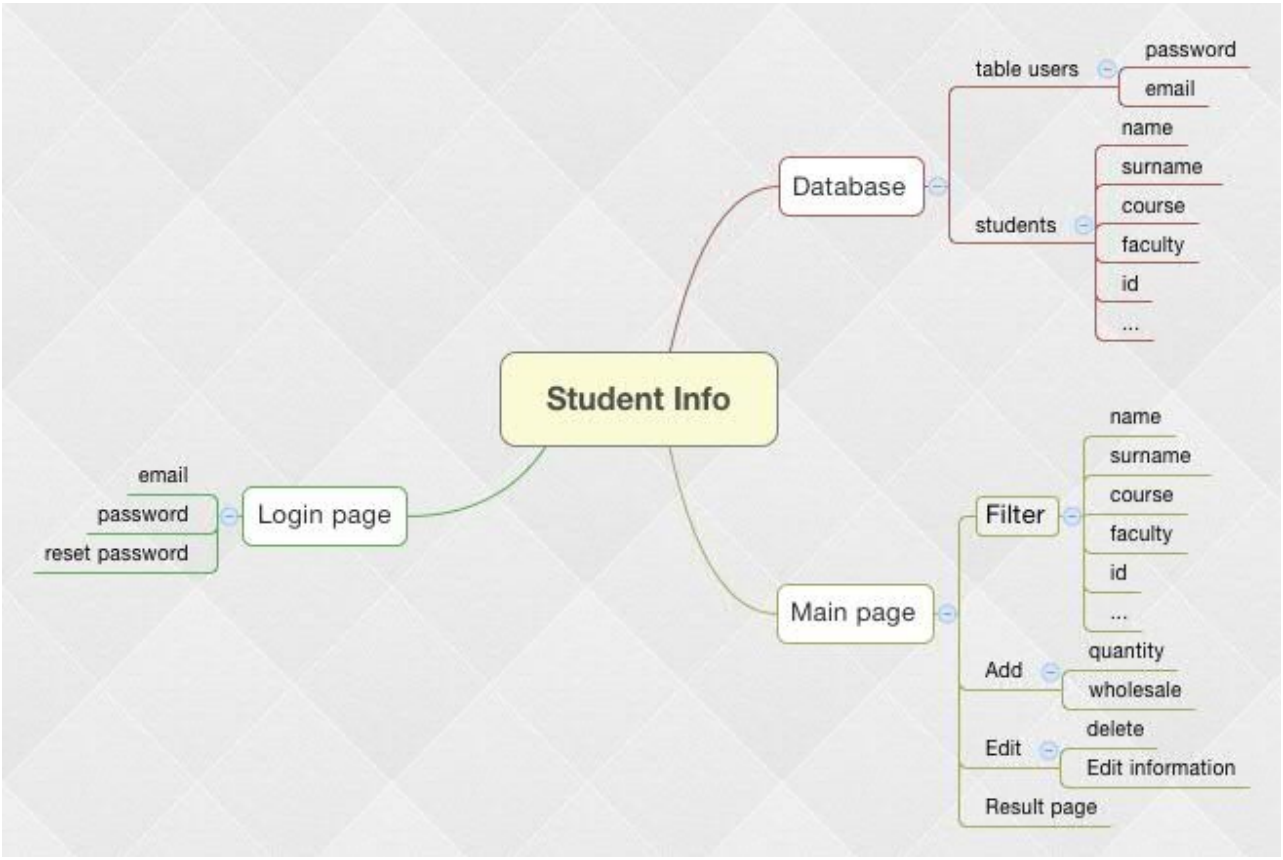
1. Application Privacy: All search results must be serialized after search query
2. Maintainability: Our code must be well organized with consistent syntax, with meaningful naming techniques and by comments.
3. Extensibility: The application must facilitate easy extension of the Graph and PDF module and features to be added to the PHP Code.
4. Response Time: All pages and search query results must be load in less than 20 seconds
5. Reliability: The core process (major use-cases) must continually function consistently, and loss of users' data must be prevented
6. Visual Consistency: All colors and design must be consensual

2 References

[None]

3 Decomposition Description

3.1 MODULE DECOMPOSITION



3.1.1 Login page Description

This page is used for enter to our web application. In order to enter to our application user must enter correct email and password. If user forgets password, our application resets password of given email (in our application user uses email as login) and send notification to email with new password

3.1.2 Database Description

Database is stored to store all data. We use MySQL database in our application. In database there will be more than 10 tables. To request data from database we use ajax+php. On request, database will return corresponding data.

3.1.3 Main page Description

User gets Main page if and only if it enters correct data to Login page. From Main page user can enter to other 4-5 pages like Settings, Add, Edit, Filter, Result

3.1.4 Settings page Description

By help of this page user can edit password and email

3.1.5 Filter Description

It is not a whole page. It is like menu, some part of page. It stands at the left side of Main page. By help Filter user can easily search for some student(s) by clicking buttons, by checking checkboxes or radio buttons and by selecting some values from select element

3.1.6 Result Description

It is not a whole page. It is some part of page. It stands at the right side of Main page. This part of page shows result of filtering, searching query and graphs of returned data

3.1.7 Add page Description

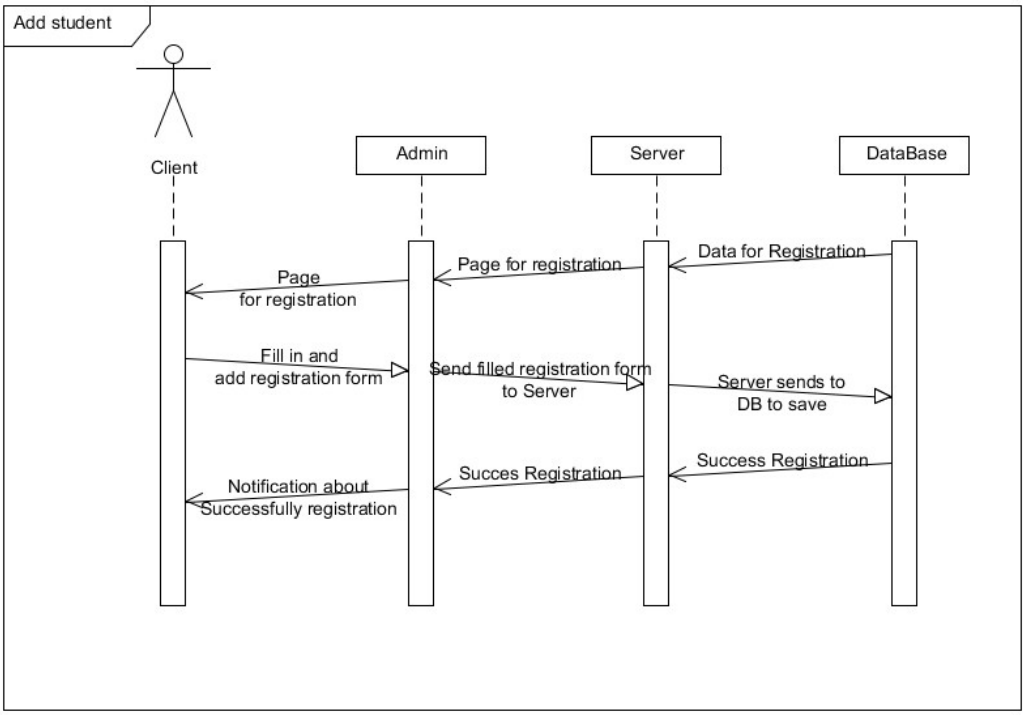
In this page user adds new student to database

3.1.8 Edit page Description

By help of this page user can edit information about selected student or remove selected student from database

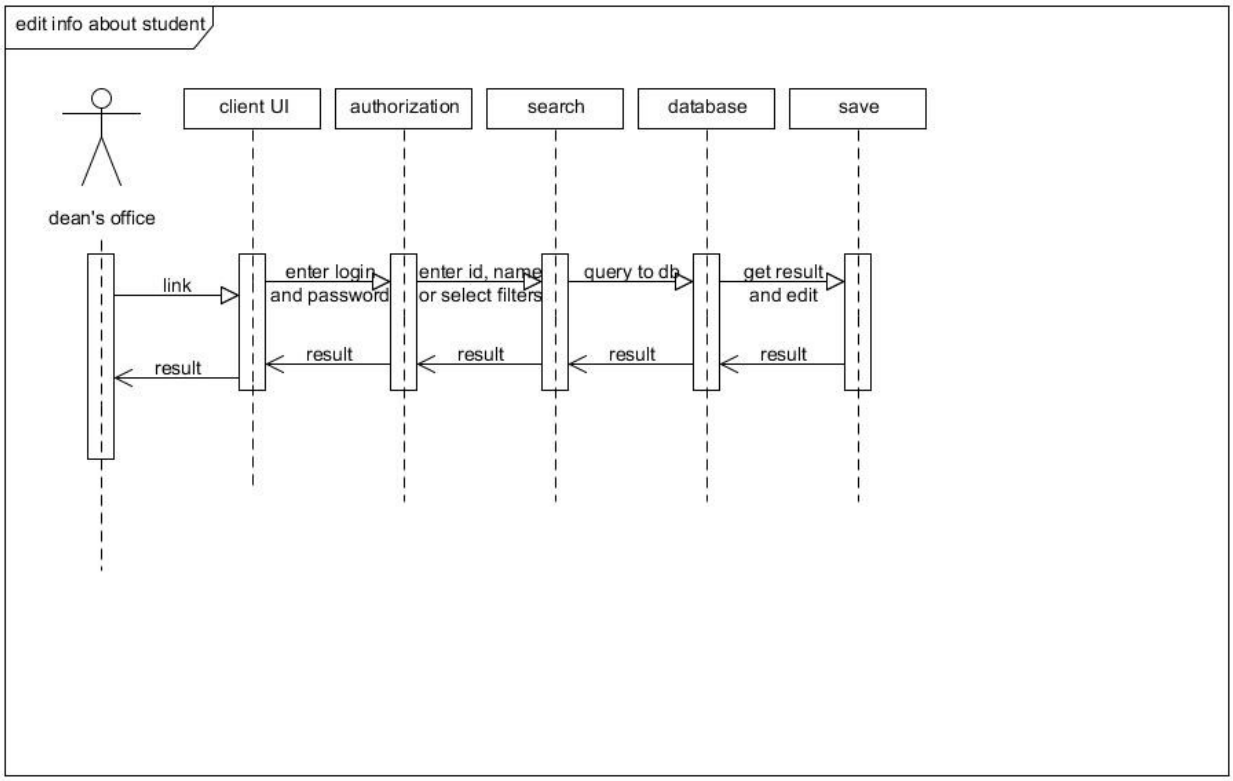
3.2 CONCURRENT PROCESS

3.2.1 Add student process Description



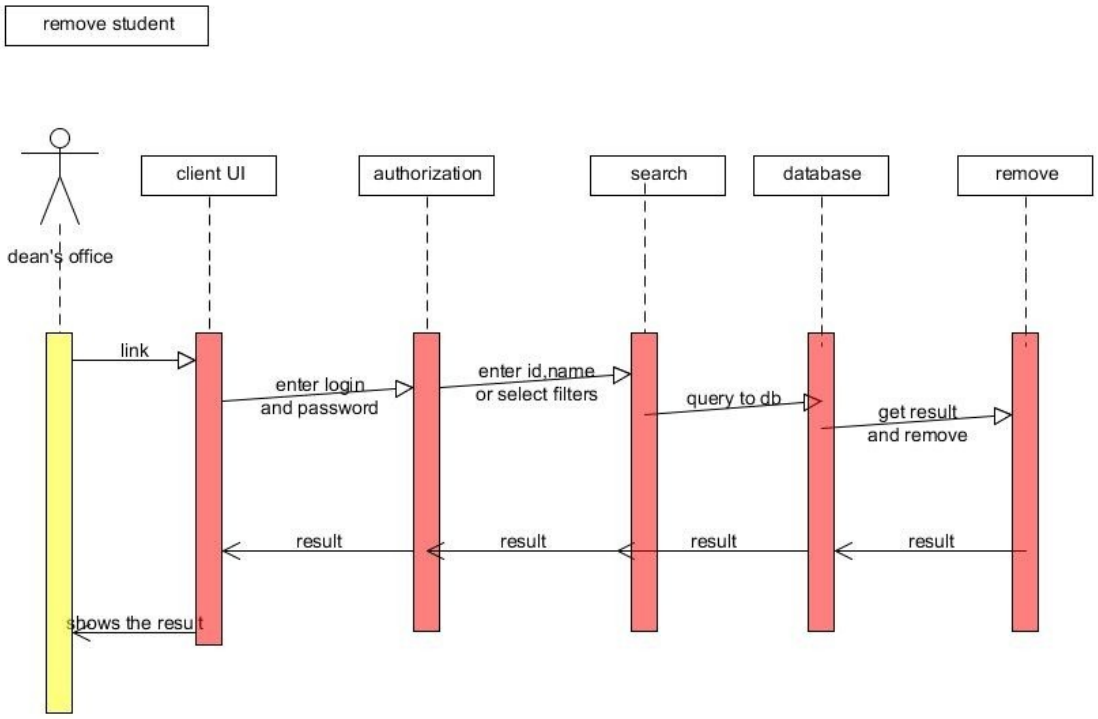
This sequence diagram shows add student pages' process. After entering to this page user need to enter data to required inputs. If error occurs like wrong data, application will notify user by notification messages. After filling inputs user need to click save button in order to send this information to database. If everything is ok, data will be store to database

3.2.2 Edit student process Description



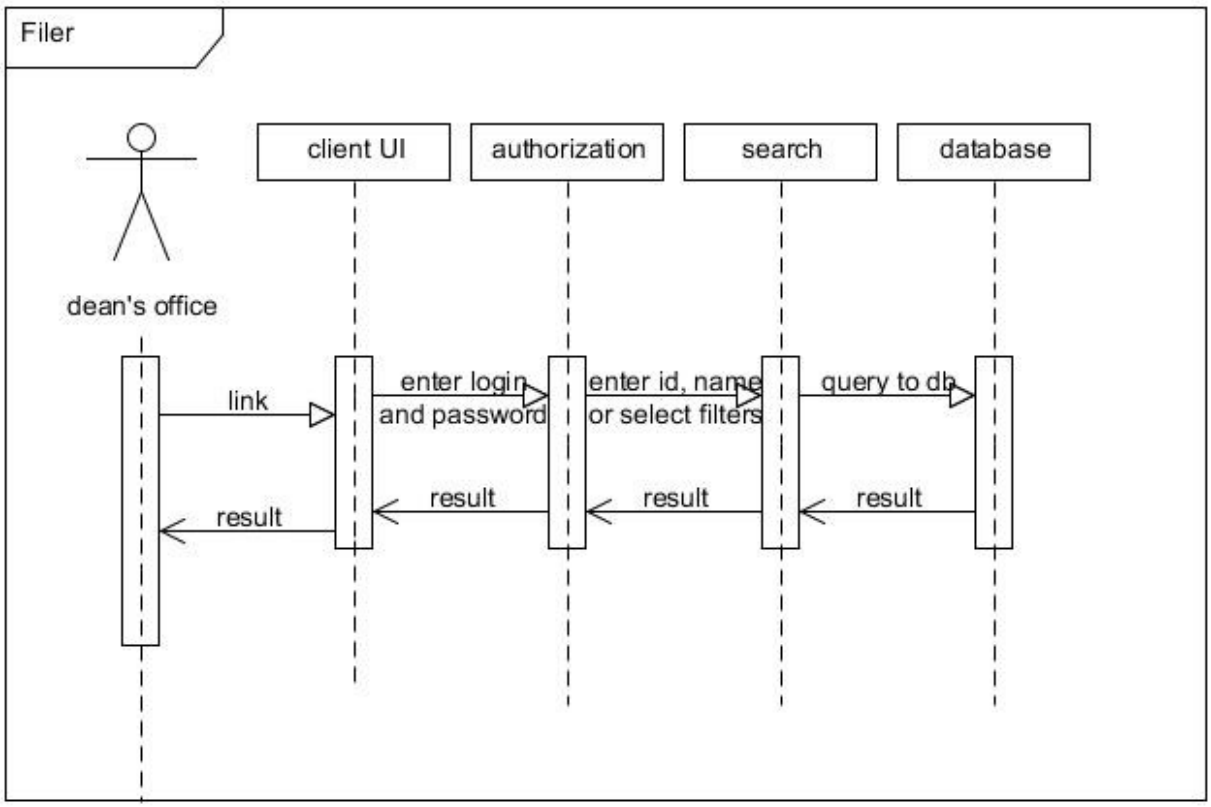
This sequence diagram shows edit student pages' process. In order to edit student user need to find this student by help search. Then user need to click to student. When user clicks to student application sends query for this student to student. Then Edit student page will open. Edit page gets data from query result and shows to user in logical order. After user can edit student and click button save in order to save changes. When save button clicked application send this data to database. If everything is ok, it updates information about selected student.

3.2.3 Remove student process Description



This sequence diagram shows remove student pages' process. In order to remove student user need to find this student by help search. Then user need to click to student. When user clicks to student application sends query for this student to student. Then Edit student page will open. Edit page gets data from query result and shows to user in logical order. After at the end of page there will be to buttons save and remove. To remove student user need to click remove button. When remove button enters application sends query to database to remove student. After removing user get Main page and notification about the result of process.

3.2.4 Filter process Description



This sequence diagram shows filtering process. On main page in the left side there will be search section. In this section there are checkboxes, buttons, select elements and radio buttons which helps to simplify and decrease search time. For example when user clicks 'male' radio button and selects 'I' from course select element, application sends query to database to get all male students whose studies on Ist course. Result section will show this query result as table.

3.3 DATA DECOMPOSITION

3.3.1 Student database table Description

'id' – id of student on database, used for queries
'sdu_id' – id of student on SDU
'name_kz' – name of student [kazakh language]
'surname_kz' – surname of student [kazakh language]
'fathername_kz' – father name of student [kazakh language]
'name_en' – name of student [english language]
'surname_en' – surname of student [english language]
'gender' – gender of student
'birthday' – datetime, day month and year of birth
'home_address_id' – home address of student, takes data from address table (3.3.2)
'current_address_id' – current address of student, takes data from address table (3.3.2)
'parents_id' – information about parents of student, takes data from parents table (3.3.3)
'faculty' – faculty of student
'department' – department of student
'course' – course of student
'group' – group of student
'gpa' – gpa of student
'email' – email of student
'phone_no' – phone number of student
'grant_type' – grant type of student (ex: sdu grant)
'stipend' – gives information about has student stipend or not
……..

3.3.2 Address database table Description

'id' – id of address, used for queries
'republic' – republic (ex: Kazakhstan)
'city' – city (ex: Taraz)
'region' – region (ex: South Kazakhstan)
'addr' – street and number of home (ex: Abay 151)
'no' – phone number of home (ex: 333-33-33)

3.3.3 Parents database table Description

'id' – id of parents, used for queries
'father' – name and surname of father
'mother' – name and surname of mother
'opekun' – name and surname of opekun
'father_phone_no' – phone number of father
'mother_phone_no' – phone number of mother
'opekun_phone_no' – phone number of mother
'father_work' – information about fathers' work
'mother_work' – information about mothers' work
'opekun_phone_no' – phone number of opekun

3.3.4 Family member database table Description

'id' – id of member, used for queries
'student_id' – student id from student table (3.3.1)
'name' – name of member
'surname' – surname of member
'type_of_affinity' – type of affinity (ex: sister)
'birthday' – datetime, day month and year of birth
'study_info' – information about university or school member (if studies)
'work_info' – information about work of member (if studies)

3.3.5 User database table Description

'id' – id of user, used for queries
'email' – email of user, used to login to application and reset password
'password' – encrypted password

3.4 STATES

3.4.1 <State/System 1 > Description

3.4.2 <State/System 2> Description

4 Dependency Description

4.1 INTERMODULE DEPENDENCIES

4.2 INTERPROCESS DEPENDENCIES

4.3 DATA DEPENDENCIES

5 Interface Description

5.1 MODULE INTERFACE

5.1.1 <Module 1> Interface

5.1.2 <Module 2> Interface

5.2 PROCESS INTERFACE

5.2.1 <Process 1> Interface

5.2.2 <Process 2> Interface

6 Detailed Design

NOT REQUIRED <Java Docs to be used instead>

7 Design Rationale

7.1 DESIGN ISSUES

7.2 <ISSUE 1>

7.2.1 Description

7.2.2 Factors affecting Issue

7.2.3 Alternatives and their pros and cons

7.2.4 Resolution of Issue

7.3 <ISSUE 1>

7.3.1 Description

7.3.2 Factors affecting Issue

7.3.3 Alternatives and their pros and cons

7.3.4 Resolution of Issue

8 Traceability

No	Use Case/ Non-functional Description	Subsystem/Module/classes that handles it
1		
2		

FEEL FREE TO ADD APPENDICES AS NEEDED. UPDATE TOC BEFORE SUBMITTING

Minutes of Meeting Form					
Team #: 2 Date: 20.03.2015					
	Student Name (initials)	Present?	Late > 5 minutes?	Informed about absence?	Scribe?
1	BS	yes	no	----------	----------
2	DM	yes	no	----------	----------
3	BA	yes	no	----------	----------
4	ZA	yes	no	----------	----------
5	TA	yes	no	----------	----------

	Student Name (initials)	Old Action Item	Status
1	BS	SRS-2 3.4	done
2	DM	SRS-2 3.2	done
3	BA	SRS-2 3.3	done
4	ZA	SRS-2 3.1	done
5	TA	SRS-2 3.5	done

Agenda / Discussion Summary
Below new action items for dd-1.

	Student Name (initials)	New Action Item	Due Date
1	BS	Team work distribution form, minutes of meeting, 3.2 3.3	21.03.2015
2	DM	1.1 1.2 1.3	21.03.2015
3	BA	1.4 2	21.03.2015
4	ZA	3.1.*	21.03.2015
5	TA	3.1.*	21.03.2015

Team Work Distribution Form					
Assignment #: 5 **Team #: 2** **Date: 21.03.2015**					
	Student Name (initials)	Signature	% of total effort (adds to 100)	Lots of extra work?	Description of what done
1	BS	----------	90	no	Team work distribution form, minutes of meeting, 3.2 3.3
2	DM	----------	85	no	1.1 1.2 1.3
3	BA	----------	75	no	1.4 2
4	ZA	----------	70	no	3.1.*
5	TA	----------	70	no	3.1.*