**Student Registration System**

Software Design Document

Rafat Munshi

Aysha Khan

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **1.** | 1.1 | **INTRODUCTION**  Purpose | **3**  3 |
|  | 1.2 | Scope | 3 |
|  | 1.3 | Overview | 3 |
|  | 1.4 | Reference Material | 3 |
|  | 1.5 | Definitions and Acronyms | 3 |
| **2.** |  | **SYSTEM OVERVIEW** | 3  **3** |
| **3.** |  | **SYSTEM ARCHITECTURE** | **3** |
|  | 3.1 | Architectural Design | 4 |
|  | 3.2 | Decomposition Description | 5 |
|  | 3.3 | Process Analyses | 12 |
| 4 | 4.1 | Dependency Description Interprocess Dependancy  Inter process dependancy |  |
|  | 4.2 | Data dependancy | 13 |
|  |  |  |  |
| **5.** |  | **HUMAN INTERFACE DESIGN** | **13** |
|  | 5.1 | Overview of User Interface | 13 |
|  | 5.2 | Database ER Diagrams | 19 |
|  | 5.3 | Details of DB tables | 19 |
| **6.** |  | **UI and Functional test cases** | **20** |
|  |  |  |  |

**1. INTRODUCTION**

**1.1 Purpose**

This document outlines the Software Design Specifications as part of the design plan and specifications for developing a web based application for Student Registration System for AMU.

This document expands the functionality described by the features in the Software Requirements Specifications (SRS). Each feature discussed will describe the variables, database query and methods to be implemented.

**1.2 Scope**

This document takes the features as outlined in the SRS and expands each of the features to include the design issues of user interface, data flow, process analysis and then module design. The design document with take a Feature based approach to the design document. The structure is discussed and described in the Entity Relationship Diagram(ERD) for the main Use-Case in the Software Requirement Specifications(SRS). The interfaces are illustrated in the dataflow and the sequence diagrams. The Graphical User Interfaces are discussed and illustrated with mock ups of the panels to be implemented. The application data is addressed in the discussion of the dataflow diagrams.

**1.4 Reference Material**

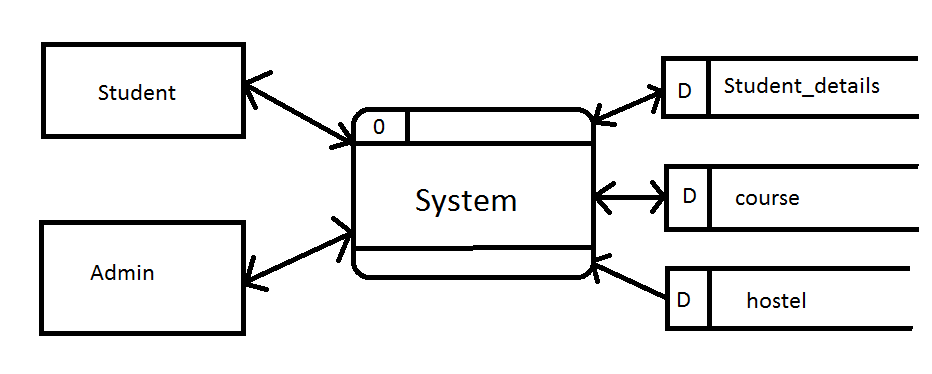
*IEEE Recommended Practice for Software Design Descriptions*, IEEE Std 1016-1998 (Revision of IEEE Std 1016-1987) Available at http://web.nps.navy.mil/~nschneid/is3020/PDF/1016-1998.pdf

**2. SYSTEM OVERVIEW**

Student Registration System is a web based application built with php and mysql database for students to register in a course in a particular semester of the university and view and edit their details. An administrator of the system can retrieve data about all students in the course semesters and view and update any student details as well as remove student/s from the system database.

**3. SYSTEM ARCHITECTURE**

**3.1 Architectural Design**

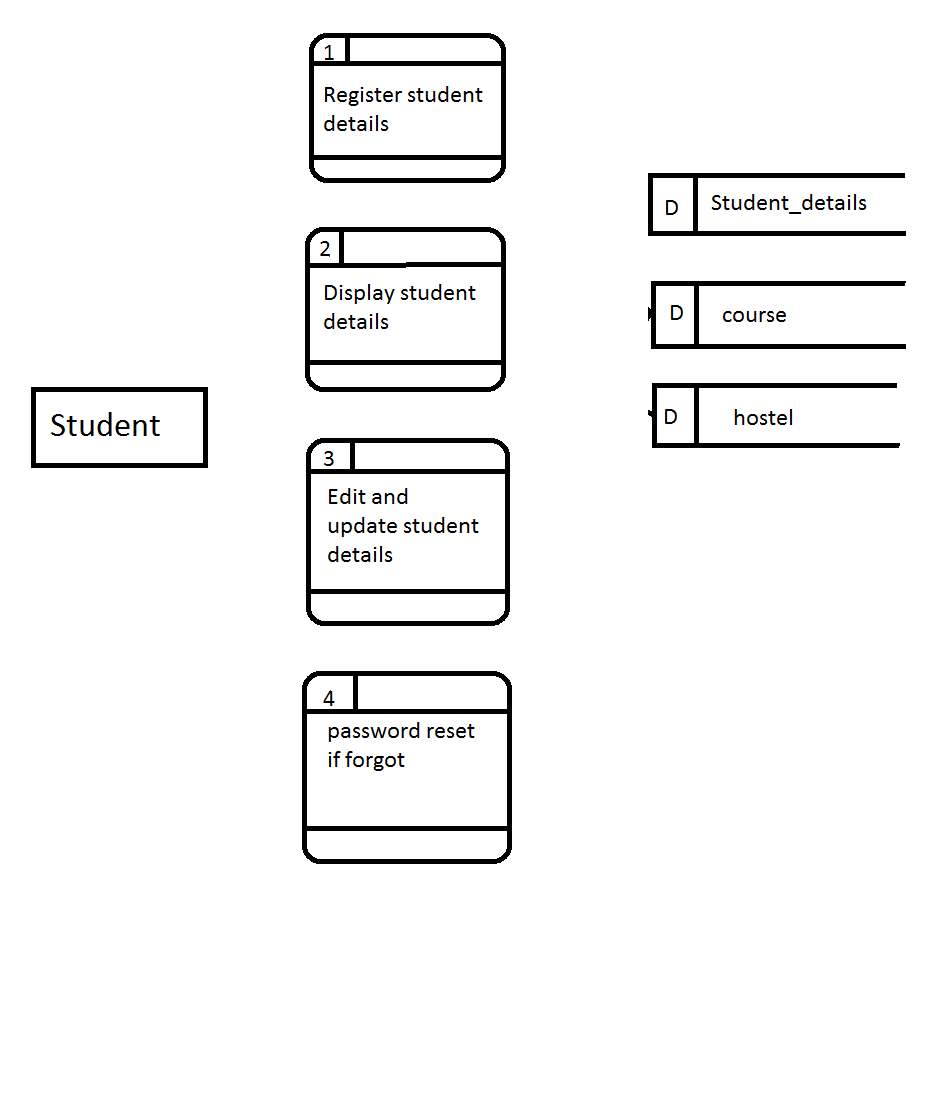
****

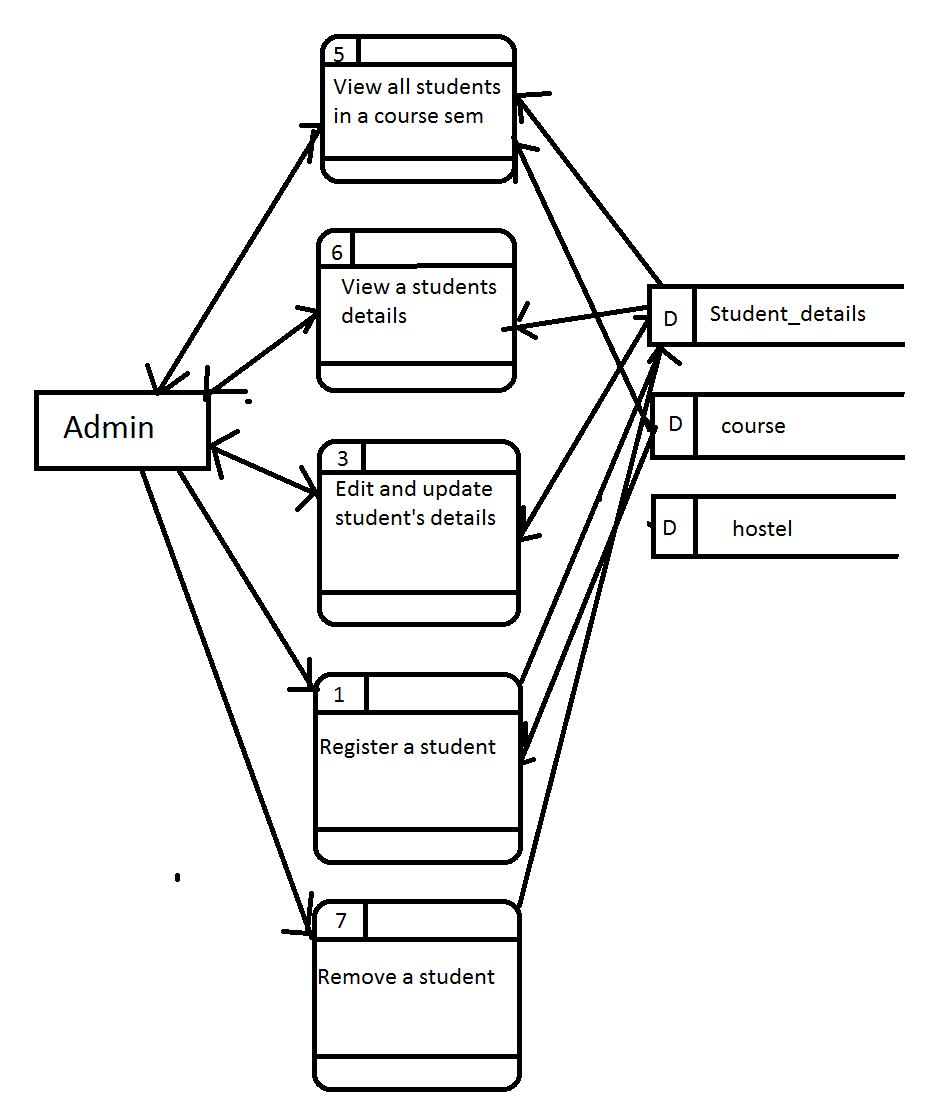
**3.2 Decomposition Description**

This section of this report decomposes use-case features as provided in the SRS document into its data flow processes by examining its data flow diagram and its process flow through the use of sequence diagrams.

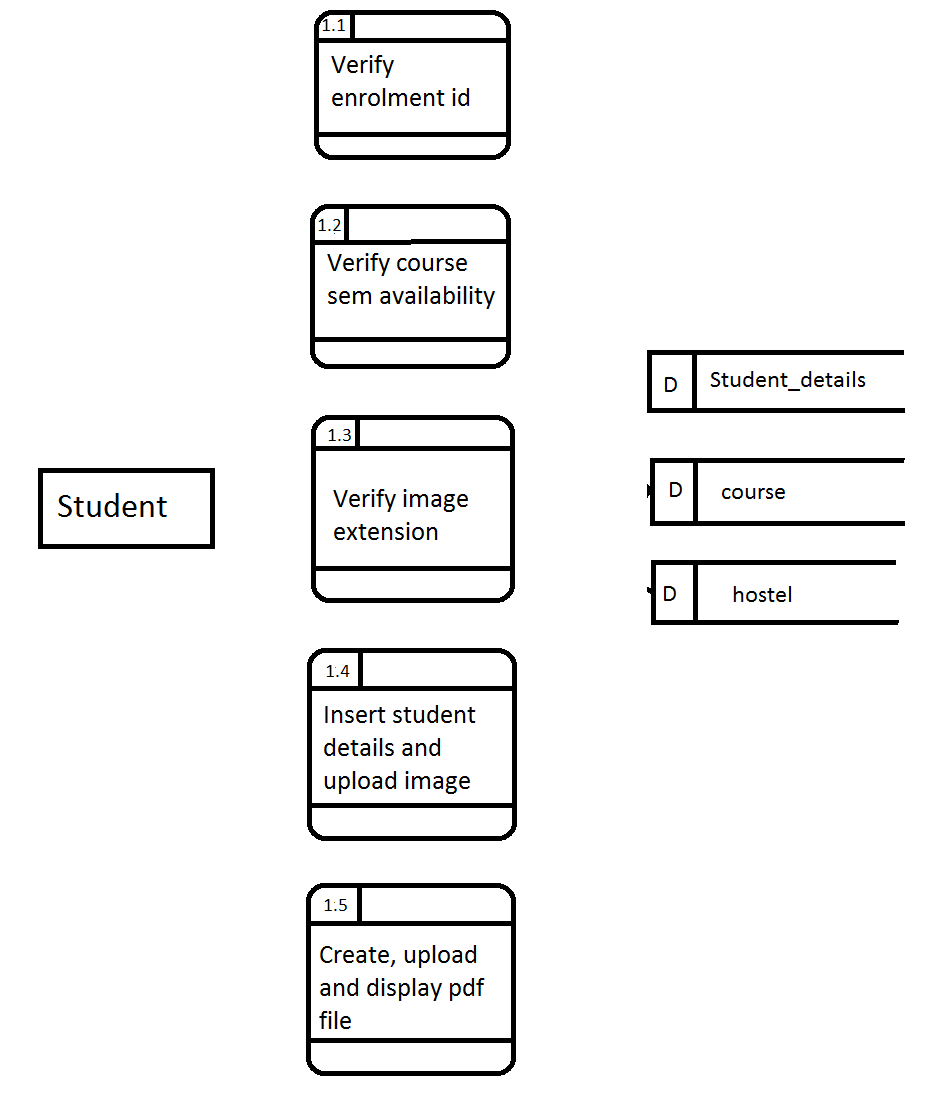
### 3.5.2. Data Flow Description

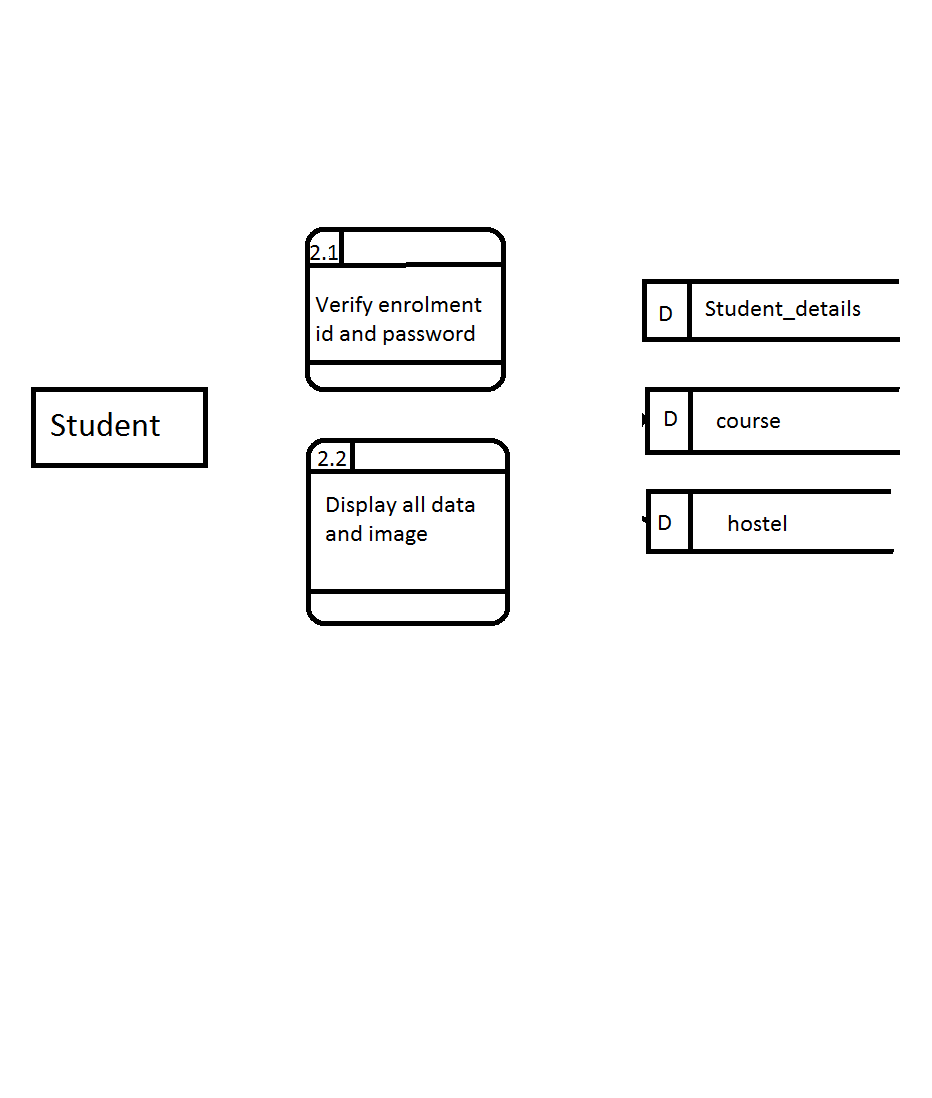
DFD level 1

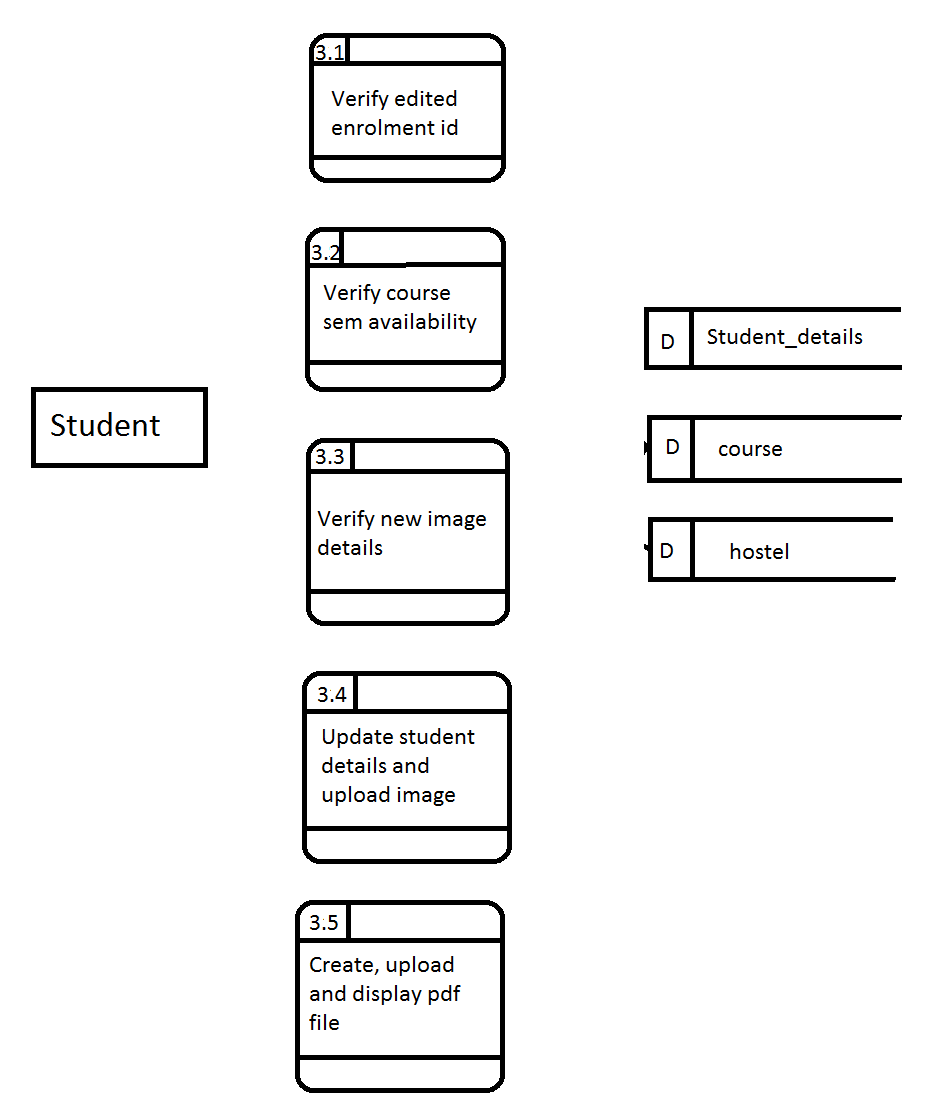


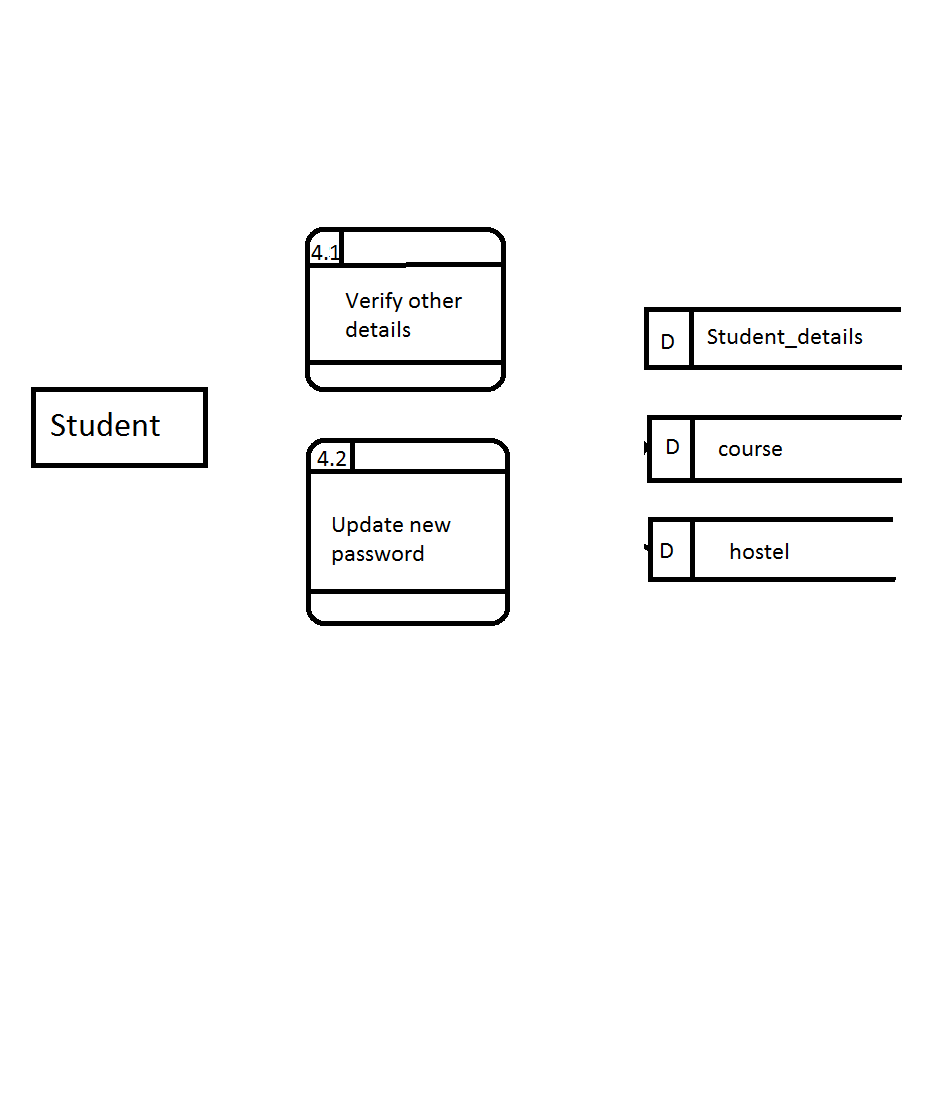


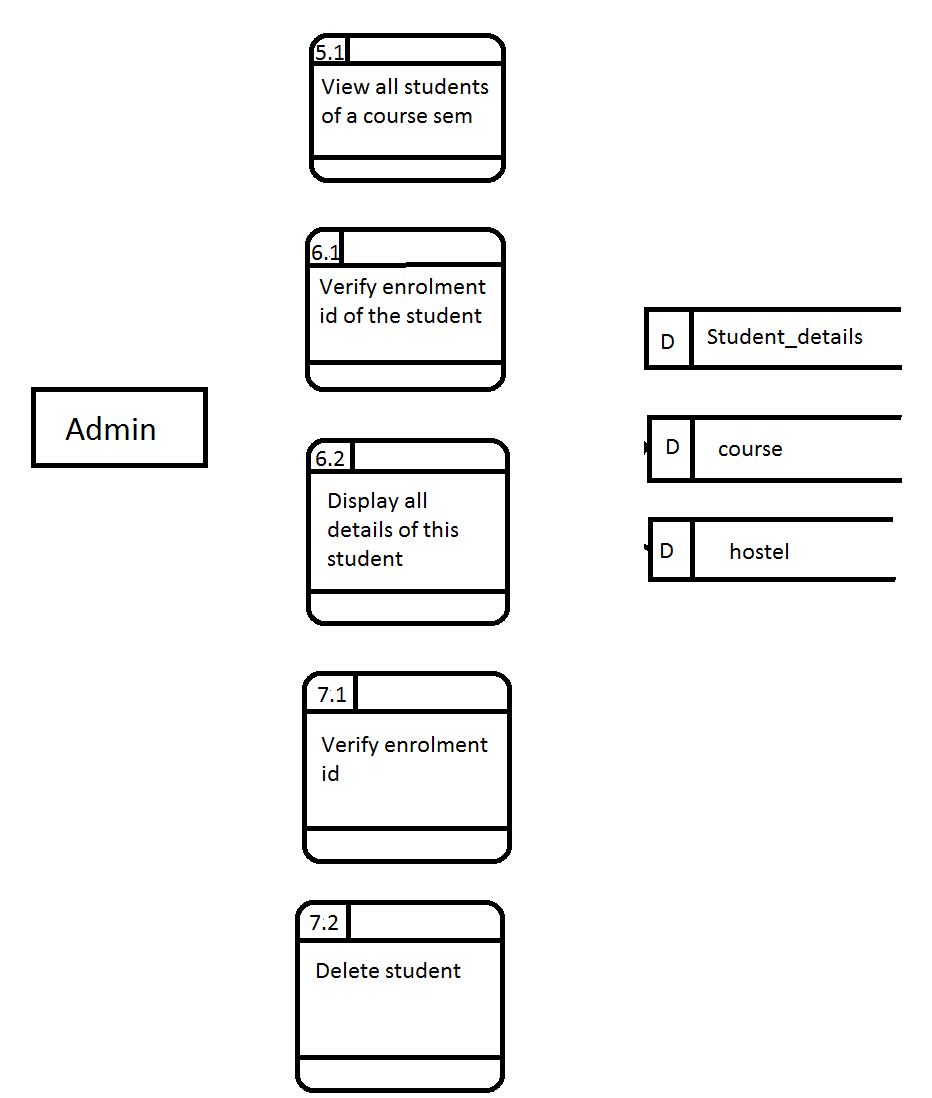
DFD level 2











### 3.3. Process Analyses

The following sequence diagram shows an actual process involved in registering new student.

Sequence diagram for registering student

### 4. Dependency Description

## 4.1 Interprocess Dependencies

The interprocess dependencies are illustrated in the data flow diagrams of the features. These will be reviewed and incorporated within the same directory of files.

## 4.2 Data Dependencies

The data dependencies are very limited in this design.

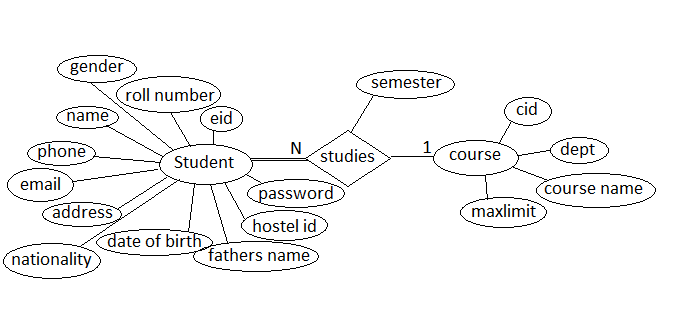
**5. HUMAN INTERFACE DESIGN**

**5.1 Overview of all the files required**

Based on the data flow diagram and the sequence diagram above, a detailed description of the required files and their description along with their UI mockup is shown below.

|  |  |  |
| --- | --- | --- |
| File name | Description | Mock up |
| home\_page.php | The UI is supposed to be created according to this mock up with html5, css3 and jquery library of javascript. It should take inputs as shown in the figure. It should contain all the validation of the input fields as specified in the SRS. |  |
|  | An alert box in case enrolment id trying to be registered already exists. |  |
|  | The second tab for the student is the login box. Here the fields are required and should be validated with the database for the details of the student. If no such id and password exist then alert the user for wrong id and password. If id and password are correct then store the id and password in the session and direct to det.php. |  |
| complete\_regis.php | All the front end validations as specified in the SRS should be incorporated in this page using javascript.  Here the course and semester selected should be checked whether max limit is reached. If so an alert box should be shown with message of course semester registration is over. The student should be allowed to register in another course semester. The image should be validated for its extension and if another extension is used then alert the user the allowed extension detail and reset the field to No file chosen. Only .jpg extension images should be uploaded. The course options are populated from the database with the cid as values and course names as display names. |  |
| Upload.php | to be called on complete\_regis.php form action. The image should be uploaded in the file system of the server with the name of the enrolment id of the student. All the details should be updated in the database. |  |
|  | A pdf generator- fpdf should be used to generate and upload this pdf in the file system of the server with the name being the enrolment id of the student. |  |
| Det.php | All the details of the student should be fetched from the database and be displayed here using the session variable of enrolment id and password for student and only enrolment id for admin. All the boxes are editable. On click of update button, all details should be updated in the database and if a new file is selected as image, that should override the previous image in the directory of images. The course options are populated from the database with the cid as values and course names as display names and default value selected is the course of the student.  On click of update button upload.php should be called hence need to put in the form action of det.php |  |
| admin\_page.php | Both fields are required. The course options are populated from the database with the cid as values and course names as display names.  A fetch of all students enrolled in the course and semester of the university should be done from the database on click of submit button. |  |
|  | validate for presence of enrolment id of the student from the database. If no enrolement id as that exists then alert user that no such student exists. If it exists then get all the details of the student from the database and display on det.php. On the click of add/remove student link, there should be redirect to addremovestudent.php |  |
| Course\_details.php | The image of the student should be fetched from the file system and displayed with the other details. |  |
| Addremovestudent.php | Add student tab same as in home\_page.php. On the click of back link, there should be redirect to admin\_page.php |  |
|  | validate for presence of enrolment id of the student from the database. If no enrolement id as that exists then alert user that no such student exists. If it exists then remove the student details from the database and alert a confirmation message of student deleted to the user. On the click of back link, there should be redirect to admin\_page.php |  |
| Forgotpw.php | The student on clicking of forgot password, should be checked with the other details- guardian’s name and date of birth. If these details of student with the enrolment id is verified and true then update the password in the database, else pop up alert of records do not match. |  |

**5.2 The database ER diagram-**



**5.3 The details of the DB tables-**

Course

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Variable Type | Variable Size | Constraint |
| Cid | int | 10 | Primary key |
| Course name | Varchar | 30 |  |
| Dept | Varchar | 30 |  |
| Max Limit | Table:1.1 |  |  |

hostel

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Variable Type | Variable Size | Constraint |
| Hostel id | int | 10 | Primary |
| Hostel name | Varchar | 50 |  |

Table:1.2

Student\_details

Table:1.4

|  |  |  |  |
| --- | --- | --- | --- |
| Variable Name | Variable Type | Variable Size | Constraint |
| eid | varchar | 7 | Primary |
| rno | int | 3 |  |
| name | varchar | 30 |  |
| phone | int | 15 |  |
| email | varchar | 30 | unique |
| pic | varchar | 20 |  |
| address | varchar | 100 |  |
| dob | date |  | Table:1.2 |
| Fathername | varchar | 30 |  |
| hostelid | varchar | 10 |  |
| password | varchar | 20 |  |
| cid | int | 10 |  |
| sem | int | 2 |  |
| gender | varchar | 7 |  |
| nationality | varchar | 30 |  |

**6. UI and functional test cases-**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |
| Sr no | Test case | Description | Expected outcome | Pass/Fail | comments |
| 1 | Application startup | On going to the link-localhost/student/student\_page.php on local server on which the application is running, a student should be able to view an interface for signing up or logging in to the student registration system. | home\_page.php is displayed | PASS |  |
| 2 | Sign up | For signing up a student is able to input his/her name, mobile number, enrolment id, password and confirm password as the first phase of registration. | Sign up tab is displayed | pass |  |
| 3 | Validation Check | When entering the details the system accept the details on the basis of some validations. The user must enter only a 10 digit phone number. The password must not be less than 8 digits. The password and confirm password must be same.The details will not be accepted if any of the fields are left empty | All validations return error pop up wherever false | pass |  |
| 4 | Database update after first phase of registration. | The details entered by the student creates a new row in the student\_details table with the details. | All details of the student successfully inserted in the database | FAIL | Phone number not updating. DB warning out of range. Needed to change int to varchar to increase range |
| 5 | Redirect to completeregis.php page | After clicking "getting started" , the student is redirected to completeregis.php page for the second phase of registration. | completeregis.php is displayed | pass |  |
| 6 | Second phase of registration | The student is provided interface to input other details like image, email id, roll number, hostel id, guardians name, address, course details, semester, date of birth, nationality, gender and a checkbox to accept terms and conditions of being a student at AMU. The semesters shown should be according to the time of the year in which the registration is taking place. For june to dec, odd number of semesters and from jan to may, even number of semesters. A complete registration and a back button is provided. | Required fields in completeregis.php are present | pass |  |
| 7 | Validations on other details. | The image should be jpg format, if not an alert message with allowed extension detail is popped up. It only accepts jpg file. All fields are compulsary. | All validations result in a pop up wherever anyone is false | pass |  |
| 8 | Back button on completeregis.php | On click of back button on completeregis.php, the user is redirected back to the home page. | redirect back | pass |  |
| 9 | Completing registration | Once the student fills in his/her details, on click of complete registration, a check to the database for course availability by comparing with the maxlimit. If available, all the details will be filled in appropriate columns of the student\_details row in the database. Also a pdf for the reference of the student will be displayed with a success message and the details which the student had input. Image is uploaded. If course maxlimit is reached then alert limit reached message. | Database is successfully updated for that student. Pdf is uploaded in uploads folder with the name being enrolment id and displayed on browser. Image is uploaded in uploads | FAIL | course availability showing full due to incorrect condition |
| 10 | Login | A student after clicking the login tab of the home page is provided with an interface to enter enrolment id and password and a login button to login into the system. A forgot password link to reset password is also provided in this interface. | Login tab with required fields is displayed | pass |  |
| 11 | Login Validation | Check with the database if enrolment id and password match any record, if so redirect to det.php else alert a wrong username and password message. | Alert on wrong input and redirection to det.php on correct input | pass |  |
| 12 | Student details display | A page now displays all the details of the student in input boxes so that the student is able to edit the details if required. An update button is provided along with a back button. | det.php opens up with required details | pass |  |
| 13 | Update student details by student | On det.php, now the student can update his/her details. All the earlier client side validations and course availability check should work. On click of update, any image uploaded should overwrite the earlier one and pdf with new details is created, uploaded and displayed on the browser. | All new details updated in database and uploads folder | pass |  |
| 14 | Back button on det.php | On click of back button on completeregis.php, the user is redirected back to the home page. | redirect back | pass |  |
| 15 | Admin Login | The admin needs to login into the system through the login tab. The enrolment id- 0000 and password- amu\_reg for the admin is already fixed. | admin\_page is displayed | pass |  |
| 16 | Student enrolled in a course | on admin\_page.php, the admin can input the course and sem detail which he/she wants to view the list of students in. On submit, course\_details.php opens up with the list of students enrolled in it in a tabular form along with their image. | on submit, course\_details.php opens with the students list and images | FAIL | 1. only image of last student could be seen. Display image code was outside loop. |
| 17 | Search student details | The admin is allowed to search the details of a student through enrollment id. The various details of the student are fetched from the database if the following enrollment id is found. Otherwise a message is displayed that no student record with this eid found. | det.php with student details opens up if eid present in db else alert pops up | pass |  |
| 18 | Update student details | The admin can update the various details of the student in the database. After editing details of the students on det.php, after client side validations, and checking availability in the course, the details of the student gets udpated in the database. | new details updated | pass |  |
| 19 | Add student | On click of Add/remove student link on admin page student tab, the addremovestudent.php page is displayed with student sign up tab. | addremovestudent.php page is displayed with student sign up tab. | pass |  |
| 20 | Remove student | This allows the admin to delete the details of a particular student based on the enrollment id of the student. After the admin enters the enrollment id, if the eid exists then the particular rows corresponding to that student in the database gets deleted from the database. If the eid does not exist then an alert message pops up saying no student with this eid is registered. | student record deleted from database if eid present else error alert pops up | pass |  |
| 21 | Back button on addremovestudent.php | The back button on addremovestudent.php will take user back to main admin page. | redirect back | pass |  |
| 22 | Forgot password by student | On clicking of forgot password, a student should be redirected to forgotpw.php | Redirect to forgotpw.php | pass |  |
| 23 | Forgot password window | The user is able to input the details on the UI – guardian name, date of birth, enrolment id, set password and confirm password along with a submit button | The required fields are present | pass |  |
| 24 | Reset password | On verification with the database if all the fields are correct for the student with that eid, then update password in database and alert confirmation message else alert records do not match message. | The password is updated in db if all fields are correct else alert error | pass |  |