1. INTRODUCTION

1.1 BACKGROUND AND MOTIVATION

Today, online education is revolutionizing the world. Online education was designed to break with traditional education methods. The introduction of online education is a way for a student to go to an institution to get an education, waste his time, pay for his daily commute there, and break the stereotype of getting an education. Online education is a platform for people to consider their time and convenience to learn something. Online education can be divided into two types. One is given live under the guidance of a teacher and two are pre-recorded passages (videos, notes) that the student learns according to his time. The similarity between the two methods is that the student learns while sitting in his private space. Though there are numerous examples of the usage of machines and tools in education throughout history, e-learning in the modern sense of the term is a relatively new concept. Slide projectors and television-based classes have been in use since the 1950s. However, one of the first instances of online learning in the world can be traced back to 1960, at the University of Illinois, USA. Though the internet wasn’t invented back then, students began learning from computer terminals that were interlinked to form a network.

The first-ever completely online course was offered in 1984 by the University of Toronto. In 1986, the Electronic University Network was established for being used in DOS and Commodore 64 computers. Three years later, the University of Phoenix became the first educational institution in the world to launch a wholly online collegiate institution, offering both bachelor’s and master’s degrees. This was the beginning of a revolution whose potential was largely unknown to the public back then, but one that would make learning greatly accessible and within reach of what people could ever have imagined. The Open University in Britain was one of the first universities in the world to begin online distance learning, in the early 1990s. Currently, the Indira Gandhi National Open University in India is the largest university in the world with around 4 million students enrolled, most of whom currently receive education via online methods. Online learning is booming in current times. Aided by the widespread availability of high-speed internet, making use of new technologies such as 4G and the soon-to-be-released 5G, online learning is expected to grow by leaps and bounds in the foreseeable future. The worldwide market size of online learning is approximately $187.87 billion in 2019, a 400% increase over what it was just six years ago. This phenomenal growth has been made possible not just by the rapidly evolving scenario in the world of technology, but also by the spread of education in the developing world. Experts predict that the next wave of online education will occur not in North America and Europe, but newly emerging markets like Africa, India, and China. Online learning is no longer just limited to colleges and universities. Right since primary school, online learning is gradually being incorporated into the curriculum. The recent COVID-19 pandemic further illustrates the importance of online learning in today’s school system, as it has proven to be a boon to both students and teachers alike who are unable to attend school due to the risk of disease spread. Beyond high school, online learning is steadily increasing its market share at the pre-university level. Furthermore, e-learning is expanding in presence beyond the traditional fields as well.

Online learning has evolved far beyond its original capabilities. It is no longer limited to a didactic method, which had a one-way monologue from the teacher to the student. Current advances in online learning enable the student to play an active role in the learning process, with regular feedback and assessments. This has greatly improved the effectiveness of the teaching system, bringing it on par with classroom-based learning. Online learning has different advantages. Less expensive than traditional teaching methods: As the cost of teaching is low, the expenses borne by the students inevitably come down. This makes education far more widespread and economical. Vast variety of available courses: These days, online courses on everything are available at the touch of a button – from religion to commerce, philosophy to fashion designing, programming to painting, photography to yoga – there is hardly any field that hasn’t been touched by e-learning. Student groups: There is a scope of engaging with like-minded students across the world, sharing information and ideas. Flexibility: This can be in terms of time, money, and location. Online learning enables the student and the teacher to be present at opposite ends of the world, in different time zones, and yet have the knowledge imparted effectively. Much less infrastructure required, this is a huge incentive to the education providers, as the additional costs are largely minimized. Standardized quality, since the content available online can be evaluated and revised at any point in time, it helps maintain a reasonable standard of quality. Smartphones have played a crucial role in making online learning viable. It is rapidly gaining ground even in rural areas, bringing high-quality education, at par with the best in the world, available to the masses. Massive open online courses (known as MOOCs) are a promising new field. The *New York Time*s had declared 2012 as “the year of the MOOC”, and there has been no stopping since then. The total number of students enrolled in MOOCs has risen to about 100 million now. Innovation in smartphone technology, High-speed data access, Interactive learning models, rising number of startups in the e-learning field etc. are helped to increase the popularity of the online education.

1.2 THE PROPOSED SYSTEM

The proposed system is developed to manage the campus virtually. Through that, faculties, students etc. can easily do the academic management, education monitoring etc. The system also has an objective to provide the complete hierarchical priorities to the users of the system which is exactly similar to the existing non virtual system. In a campus, the campus management has the power to decide all the functions of that campus. In education system the tasks are divided into different HODs whose has the charge of different departments of that campus. HODs implement their agenda through the teachers and the students are at the low level of the hierarchy. Their access in the system is limited. They can only see what they need to do and their own data. There are no permissions to access other students or faculties data. The main motive for developing this system is for reduce and overcome the complexity of the existing system and develop an interface which is able to manage the functions of a educational institution or a campus virtually. Students are the users of the system. Using this system, the students are able to submit their assignments with in the due date. If they have any doubt regarding the assignment, they can add private comments to the particular teacher who assigned the assignment to the student. Teacher can view and replay to those comments privately. Student can view and download the marks, study materials etc. from this system which is uploaded by a teacher. There is another feature offered to the students are virtual complaint registration. Using this feature students are able to give complaints to teachers, HOSs, Principal according to their need. They are also able to check their complaint status which means the complaint is resolved or not resolved or under processing or anything. There is another imported feature offered by the system that is students are able to view the timetable which is updated by the HOD. They will get notifications when a teacher starts their session and they will get the meeting link to join the session with a button click.

Another stakeholder or this system is faculties. Faculties can be classified into two. First one is HODs and second one is teachers. Teachers have privileges like host meeting or a class, update attendance, create assignments, divide the students into different groups, generate reports based on different criteria like marks or attendance or groups or assignments etc. A HOD can also be a teacher too. So, the system offers all the features of teachers to the HOD and it offers some more privileges like create/update/delete timetable, assign subjects to the teachers, host department meeting/PTA meeting/public meetings etc, generate reports of teachers based on subjects/number of hours class taken etc, generate reports of students based on marks/class/semester etc. HODs and teachers can view and register complaints. They can resolve the complaints that repoted to them and they can register new complaints to the admin or HODs. The system offers only one loin form to the HODs and other teachers to avoid the feel of priority differences among the faculties. But after login the HODs will redirect to a different page with different privileges and other teachers will redirect to other page with limited privileges than the HODs.

Admin is another stakeholder of the system. Admin may or may not be the management team of the campus or institution. In this system it doesn’t offers a registration page to register their own details to other stake holders. The admin should register the details of faculties and students to avoid the fakes. Admin doesn’t need to register the user with all the related information of the faculties and students. They only need to register with some of the details like name, email, salary etc. The username and temporary password of each user will get in the registered email of the user. The username is fixed for all the users but they can change the temporary password after first login. Faculties and students are requested to complete their profile with the details like photo, contact number, address, date od birth etc. These details will send to the admin for the verification. After verification admin can either accept or reject that. The account will be activated only after a successful verification of the admin. Admin is the overall controller of the system, they are not only registering the faculties and students but also registering departments, batches, semesters and different clubs. They can generate reports of faculties based on different criteria like HODs/ Faculty/ Students/ Salary/ Departments/ Batches/ Semesters/ Clubs etc. They are able to host public meetings, HOD meeting, faculty meeting etc. Admin can view the complaints that reported to him. They can read and update the status of that complaint.

1.3. PROJECT SCOPE

1.3.1. Limitations of Existing System

- Entire campus management

The existing systems like google classrooms are providing a facility to manage the online education classroom wise. It doesn’t follow the organization hierarchy. So, the management, HOD has no role there. And also, it is difficult to manage different classes of different departments.

- Reports

  The existing system is only developed for managing the education system. It doesn’t provide a felicity to take reports of teachers or attendance report etc.

- Complexity of data

  In the existing system the all data are arranged at a single field. So, it makes more complexity in accessing data or content.

- Lack of intelligent reminders

  Reminders have to be provided at the time of the user invited to join a meeting and at the time of host started the meeting etc. If user have to assign/submit any kind of data before a deadline the system should sent remainders before the deadline.

- Fake users

  A person with an e-mail id can join in any classrooms if they know the classroom id. It increases the possibilities of joining the fake users in to the classrooms. It can be avoided by giving organization e-mail ids. But it is not possible to provide e-mail ids to all the users in terms of some organizations.

- User login

  There are some issues related to user login. The system is not providing proper user logins.

- Loss of data

  In the google classroom the data is stored in the google drive. We can access the files that shared through the system also in drive. If any kind of modifications done in the drive can also affect the files uploaded in the system.

- Duplication of data

  If we cancel the uploading of a file in mid way, the half-uploaded file will be stored in the clod storage. If we reupload the file the uploading will resume from where we stopped. But if we upload a different file the existing one doesn’t remove automatically. It leads to duplication of data and loss of storage space.

- Complaint facility

  The systems don’t provide a complaint facility to the user. They only provide an interface to report the bugs of that particular system only.

- Paid system

  The free version of the system has more limitation in terms of number of users, availability of storage space, felicities provided etc. If the user wish to upgrade their pack, it need a huge price for an annual plan.

1.3.2. Advantages of Proposed System

- Don't miss any valuable information

  Users of this system will get reminders about their deadlines and they can generate reports based on different criteria, so they can easily classify and understand the complicated data.

- The system provides direct communication with different users

  The system is offering a virtual campus management. In a campus there are a lot of discussions can be done. As well as this system provides a communication among management, HODs, teachers and students via virtual meetings and messages/comments.

- Searching is very easy

  This is very helpful to search and find there need easily and fast. If a student needs to access to the details of a particular subject they can search. If a teacher needs to find the details of student/class they can easily search it. As well as all users of the system has various search options.

- Less time consuming

  The system is developed using modern technology and by studying the existing systems limitations and complexity. In this system the complexity is reduced maximum so user can easily access to it within less time. The all process in this system is much faster so user can save more time.

- Data redundancy is avoided

  As computer is generating the reports, there won’t be any duplications or mistakes. There are validation rules and verifications to checks and prevent errors.

- Eliminate the use of paper

The system offers different functionalities which are currently done as paper works. By using the system, the user can avoid the paper works and they can take the reports any time any where by accessing their account.

- Updating details is easy

  The new system enables the users to update their information's. The Customer or the administrator can manage the datils of a user.

- The system should have new security features

  There are effective and complicated passwords in case of any accidental data loss or damage. In addition, unauthorised access of the database is prohibited. The access of fake users to this system is completely eliminated.

- The new system should be user friendly

  The new system is user friendly. The user interface of the system is attractive and simple in design. So, any new user can easily understand and can do various operations provided by the system within minutes.

- Data availability

  Data is available to each individual at any time and this provided data is well defined to be used in our education system. Availability or accessibility of data is in a very simple format. The users can use the system in any time. Customers can access the system any time and view or raise doubts at any time. (24x7)

- Data accessibility

  The proposed system is highly intellectual because user can access the system. The information is different but it is useful to us in different ways in different situations. Users can refer data to resolve any kind of complaints etc.

- Single and simple platform

  This system complains multiple systems into one system. Now we are using zoom, google meet etc. for hosting virtual meetings. We can access these systems from the proposed system by just a button click. We can also send invite notifications e-mails etc. to the users.

2.SYSTEM ANALYSIS

2.1 INTRODUCTION

Software Engineering is the analysis, design, construction, verification and management of technical or social entities. To engineer software accurately, a software engineering process must be defined. System analysis is a detailed study of the various operations performed by the system and their relationship within and module of the system. It is a structured method for solving the problems related to the development of a new system. The detailed investigation of the present system is the focal point of system analysis. This phase involves the study of parent system and identification of system objectives. Information has to be collected from all people who are affected by or who use the system. During analysis, data are collected on the variable files, decision point and transactions handled by the present system. The main aim of system is to provide the efficient and user friendly automation. So the system analysis process should be performed with extreme precision, so that an accurate picture of existing system, its disadvantages and the requirements of the new system can be obtained.

System analysis involves gathering the necessary information and using the structured tool for analysis. This includes the studying existing system and its drawback, designing a new system and conducting cost benefit analysis. System analysis is a problem solving activity that requires intensive communication between the system users and system developers. The system is studied to the minute detail and analyzed. The system is viewed as a whole and the inputs to the system are identified. The outputs from the organization are traced through various phases of processing of inputs.

There are a number of different approaches to system analysis. When a computer based information system is developed, systems analysis (according to the Waterfall model) would constitute the following steps:

- The development of a feasibility study, involving determining whether a project is economically, technologically and operationally feasible.
- Conducting fact-finding measures, designed to ascertain the requirements of the system's end-users. These typically span interviews, questionnaires, or visual observations of work on the existing system

- Gauging how the end-users would operate the system (in terms of general experience in using computer hardware or software), what the system would be used for and so on.

Techniques such as interviews, questionnaires etc. can be used for the detailed study of these processes. The data collected by these sources must be scrutinized to arrive at a conclusion.

The conclusion is an understanding of how the system functions. This system is called the Existing System. The Existing system is then subjected to close observation and the problem areas are identified. The designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as a proposal which is the Proposed System. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is then presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is a loop that ends as soon as the user is satisfied with the proposal.

2.2 STAKE HOLDERS OF THIS PROJECT

2.2.1 Admin

The admin is a person who controls all the functionality and provide access to the users. He has the main control over the shop. The admin is capable of registering faculties, students, departments, courses, batches, semesters, clubs etc... Admin can access the details of faculties, students etc... Admin can host meeting. Admin needs to provide a link of their meeting to redirect to that meeting by just a button click. Admin can select the audiences and sent a notification to them to inform that he/she started a meeting. Admin is capable of selecting a head of the department for each department from the faculties. They can monitor each of the other stake holders of this system. Admin is also capable of generating various reports based on salary of the faculties, mark of students etc... The day-to-day assessment is carried out by him.

2.2.2 HOD

A HOD is also a main stakeholder of the system. The admin is capable of selecting a HOD from the faculties. Once the HOD is registered the HOD will be able to login later to the site using their login credentials. But they need to change their one-time password provided by the admin and need to update their profile with mandatory details to use the full functionalities of the system. Admin will verify these details after that admin will either accept or reject those details based on the errors in it. If admin rejects it as the details are not clear or wrong HOD needs to resubmit their details on profile section to access the full functions of the system. HOD can create different classes of different students. HOD can register subjects and also can assign each subject to the teachers. HOD capable to host meetings by just a button click. On that click they will redirected to their provided meeting hosting software and the target audiences will notified when you starts that meeting. HOD capable to generate the timetable for each class. They have an option to submit the attendance of the faculties to this system. Later they can generate reports of the attendance of the faculties for further processes. HOD can access to the details of faculties, students who belongs to that particular department. HOD is capable of generating reports of faculties, students and subjects etc...

2.2.3 Teacher

Teachers are also the main part of campus management system. Teachers can login into the site using their login credentials. But they need to change their one-time password provided by the admin and need to update their profile with mandatory details to use the full functionalities of the system. Admin will verify these details after that admin will either accept or reject those details based on the errors in it. If admin rejects it as the details are not clear or wrong teacher needs to resubmit their details on profile section to access the full functions of the system. Teachers can assign assignments, create groups etc... Teachers are able to start a meeting from this system by just an button click. At that time a notification will be sent to the targeted students. They have an option to submit the attendance of the students to this system. Later they can generate reports of the attendance of the students, subjects for further processes. They can generate reports based on the marks, groups, submitted assignments etc. of students.

2.2.4 Student

Students are also the main part of campus management system. Students can login into the site using their login credentials. But they need to change their one-time password provided by the admin and need to update their profile with mandatory details to use the full functionalities of the system. Admin will verify these details after that admin will either accept or reject those details based on the errors in it. If admin rejects it as the details are not clear or wrong student needs to resubmit their details on profile section to access the full functions of the system. Students are assigned to a class. And they can access the data related to that class like semester, subjects, assigned assignments, notes, marks etc... They can submit their assignments and they will get notifications when a teacher hosts a meeting. Students are able to view their categorized published marks any time.

2.3 SOFTWARE REQUIREMENT SPECIFICATION

2.3.1 Admin

1. Admin Dashboard
2. Admin can add district and places
3. Admin can add departments and faculties
4. Admin can add course type and courses
5. Admin can add batches
6. Admin can add semesters
7. Admin can add clubs
8. Admin can host meetings
9. Admin can view complaints
10. Admin can view reports

2.3.2 HOD

1. HOD dashboard.
2. The HOD should have the provision to log in using username and password.
3. HOD can add classes
4. HOD can add subjects and assign faculties
5. HOD can create timetable
6. HOS can host meetings
7. HOD can view faculty details, student details etc.
8. HOD can change their dashboard to teacher mode
9. HOD can view various reports also

2.3.3 Teachers

1. Teachers also have dashboard.
2. The Teachers should have the provision to log in using username and password.
3. Teachers can add notes, marks and attendance.

4. Teacher can add groups
5. Teachers can view the assignment status.
6. Teachers can host meetings
7. Teacher can view various reports also.

2.3.4 Students

1. Students also have dashboard.
2. The students should have the provision to log in using username and password.
3. Students can view and download marks, notes and attendance
4. Students can submit assignments
5. Students can view semesters and subjects

Table 2.1. Sign off table

Sl. No.	**Name & Designation**	**Date**	**Accepted (Yes/No)**
1	Nabeel Latheef Developer		

b. FEASIBILITY STUDY

Feasibility is defined as the practical extent to which a project can be performed successfully. To evaluate feasibility, a feasibility study is performed, which determines whether the solution considered to accomplish the requirements is practical and workable in the software. Information such as resource availability, cost estimation for software development, benefits of the software to the organization after it is developed and cost to be incurred on its maintenance are considered during the feasibility study. The objective of the feasibility study is to establish the reasons for developing the software that is acceptable to users, adaptable to change and conformable to established standards. Various other objectives of feasibility study are listed below.

- To analyse whether the software will meet organizational requirements.
- To determine whether the software can be implemented using the current technology and within the specified budget and schedule.
- To determine whether the software can be integrated with other existing software.

When my project guide as well as my client Gibin George told me regarding the main project and about Word to the Wise for getting the desired product developed, it comes up with rough idea about what all functions the software must perform and which all features are expected from the software.

Referencing to this information, I do a studies and discussions about whether the desired system and its functionality are feasible to develop and the output of this phase is a feasibility study report that should contained adequate comments and recommendations.

Various types of feasibility that I checked include technical feasibility, operational feasibility, and economic feasibility.

Technical Feasibility

Technical feasibility assesses the current resources (such as hardware and software) and technology, which are required to accomplish user requirements in the software within the allocated time and budget. For this, the software development team ascertains whether the current resources and technology can be upgraded or added in the software to accomplish specified user requirements. Technical feasibility also performs the following tasks.

- Analyses the technical skills and capabilities of the software development team members.
- Determines whether the relevant technology is stable and established.
- Ascertains that the technology chosen for software development has a large number of users so that they can be consulted when problems arise or improvements are required.

From my perspective there are two languages Angular, HTML and database MySQL which are used to develop this web-based applications. ANGULAR is used in the front end and MySQL is used in the back end. The Word to the Wise is web based and thus can be accessed through any browsers. As we are using these latest technologies which are currently trending and used by a number of developers across the globe, we can say that my project is technically feasible.

Operational Feasibility

Operational feasibility assesses the extent to which the required software performs a series of steps to solve business problems and user requirements. This feasibility is dependent on human resources (software development team) and involves visualizing whether the software will operate after it is developed and be operative once it is installed. Operational feasibility also performs the following tasks.

- Determines whether the problems anticipated in user requirements are of high priority.
- Determines whether the solution suggested by the software development team is acceptable.
- Analyses whether users will adapt to a new software.
- Determines whether the organization is satisfied by the alternative solutions proposed by the software development team.

I found that my project will be satisfied for the client since we were discussing every detail about the software with the client at every step. The most important part of operational feasibility study is the input from client. So the software is built completely according to the requirements of the client. We have used the current industry standards for the software. Hence we can say that this software is operationally feasible.

Economic Feasibility

Economic feasibility determines whether the required software is capable of generating financial gains for an organization. It involves the cost incurred on the software development team, estimated cost of hardware and software, cost of performing feasibility study, and so on. For this, it is essential to consider expenses made on purchases (such as hardware purchase) and activities required to carry out software development. In addition, it is necessary to consider the benefits that can be achieved by developing the software. Software is said to be economically feasible if it focuses on the issues listed below.

- Cost incurred on software development to produce long-term gains for an organization.
- Cost required to conduct full software investigation (such as requirements elicitation and requirements analysis).
- Cost of hardware, software, development team, and training.

It is estimated that my project is economically feasible as development cost is very minimal since the tools and technologies used are available online. It’s a group student project so there are no personnel costs. Development time is well planned and will not affect other operations and activities of the individuals. Once the system has been developed, the companies purchasing the system will be providing with a manual for training purposes. There is no need to purchase new hardware since the existing computers can still be used to implement the new system.

c. SOFTWARE DEVELOPMENT LIFECYCLE MODEL

One of the basic notions of the software development process is SDLC models which stand for Software Development Life Cycle models. SDLC – is a continuous process, which starts from the moment, when it's made a decision to launch the project, and it ends at the moment of its full remove from the exploitation. Software development lifecycle (SDLC) is a framework that defines the steps involved in the development of software. It covers the detailed plan for building, deploying and maintaining the software. SDLC defines the complete cycle of development i.e. all the tasks involved in gathering a requirement for the maintenance of a Product.

Some of the common SDLC models are Waterfall Model, V-Shaped Model, Prototype Model, Spiral Model, Iterative Incremental Model, Big Bang Model, Agile Model. We used Agile Model for my Project.

Agile Model

Agile Model is a combination of the Iterative and incremental model. This model focuses more on flexibility while developing a product rather than on the requirement. In the agile methodology after every development iteration, the client is able to see the result and understand if he is satisfied with it or he is not. Extreme programming is one of the practical uses of the agile model. The basis of this model consists of short meetings where we can review my project. In Agile, a product is broken into small incremental builds. It is not developed as a complete product in one go. At the end of each sprint, the project guide verifies the product and after his approval, it is finalised. Client feedback is taken for improvement and his suggestions and enhancement are worked on in the next sprint. Testing is done in each sprint to minimize the risk of any failures.

Advantages of Agile Model:

- It allows more flexibility to adapt to the changes.
- The new feature can be added easily.
- Customer satisfaction as the feedback and suggestions are taken at every stage.

- Risks are minimized thanks to the flexible change process

Disadvantages:

- Lack of documentation.
- If a customer is not clear about how exactly they want the product to be, then the project would fail.
- With all the corrections and changes there is possibility that the project will exceed expected time

d. HARDWARE AND SOFTWARE REQUIREMENTS.

i. Software Specification

This project is built upon the latest technology software.

Front end	:	Angular, HTML, JavaScript
Development tool	:	Node, Express
Database	:	My SQL
Web server	:	WAMP server
Operating System	:	Windows 10

2.6.1.1. Node

Node.js is an open-source, cross-platform, back-end JavaScript runtime environment that runs on the V8 engine and executes JavaScript code outside a web browser. Node.js lets developers use JavaScript to write command line tools and for server-side scripting—running scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. Consequently, Node.js represents a "JavaScript everywhere" paradigm,[6] unifying web-application development around a single programming language, rather than different languages for server-side and client-side scripts.

Node.js has an event-driven architecture capable of asynchronous I/O. These design choices aim to optimize throughput and scalability in web applications with many input/output operations, as well as for real-time Web applications (e.g., real-time communication programs and browser games).

The Node.js distributed development project was previously governed by the Node.js Foundation,[8] and has now merged with the JS Foundation to form the Open JS Foundation, which is facilitated by the Linux Foundation's Collaborative Projects program.

Node.js allows the creation of Web servers and networking tools using JavaScript and a collection of "modules" that handle various core functionalities. Modules are provided for file system I/O, networking (DNS, HTTP, TCP, TLS/SSL, or UDP), binary data (buffers), cryptography functions, data streams, and other core functions. Node.js's modules use an API designed to reduce the complexity of writing server applications. JavaScript is the only language that Node.js supports natively, but many compile-to-JS languages are available. As a result, Node.js applications can be written in CoffeeScript,[50] Dart, TypeScript, ClojureScript and others.

2.6.1.2. Express

Express.js, or simply **Express**, is a back end web application framework for Node.js, released as free and open-source software under the MIT License. It is designed for building web applications and APIs.[3] It has been called the de facto standard server framework for Node.js.[4]

The original author, TJ Holowaychuk, described it as a Sinatra-inspired server,[5] meaning that it is relatively minimal with many features available as plugins. Express is the back-end component of popular development stacks like the MEAN, MERN or MEVN stack, together with the MongoDB database software and a JavaScript front-end framework or library. Express.js was founded by TJ Holowaychuk. The first release, according to Express.js's GitHub repository, was on the 22nd of May, 2010. Version 0.12

2.6.1.3. MySQL

MySQL is the world's most popular open source database software, with over 100 million copies of its software downloaded or distributed throughout it's history. With its superior speed, reliability, and ease of use, MySQL has become the preferred choice for Web, Web 2.0, SaaS, ISV, Telecom companies and forward-thinking corporate IT Managers because it eliminates the major problems associated with downtime, maintenance and administration for modern, online applications.

Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.

The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.

MySQL is a key part of LAMP (Linux, Apache, MySQL, ANGULAR / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

MySQL was originally founded and developed in Sweden by two Swedes and a Finn: David Axmark, Allan Larsson and Michael "Monty" Widenius, who had worked together since the 1980's. MySQL, the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.

MySQL is a database management system. A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

MySQL databases are relational. A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to one, one-to-many, unique, required or optional, and ―pointers‖ between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of ―MySQL‖ stands for ―Structured Query Language‖. SQL is themost common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, ―SQL-92‖ refers to the standard released in 1992, ―SQL:1999‖ refers to the standard released in 1999, and ―SQL:2003‖ refers to the current version of the standard.

We use the phrase ―the SQL standard‖ to mean the current version of the SQL Standard at any time. MySQL software is Open Source. Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs.

The MySQL software uses the GPL (GNU General Public License),http://www.fsf.org/licenses/, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us.

The MySQL Database Server is very fast, reliable, scalable, and easy to use. If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet. MySQL Server works in client/server or embedded systems.

The MySQL Database Software is a client/server system that consists of a multi- threaded SQL server that supports different backends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs). We also provide MySQL Server as an embedded multi-threaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

A large amount of contributed MySQL software is available. MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favourite application or language supports the MySQL Database Server.

2.6.1.4.WAMP Server

WAMP Server is a Windows web development environment. It allows you to create web applications with Apache2, ANGULAR and a MySQL database. Alongside, AngularMyAdmin allows you to manage easily your databases. WAMP Server refers to a software stack for the Microsoft Windows operating system, created by Romain Bourdon and consisting of the Apache web server, Open SSL for SSL support, MySQL database and ANGULAR programming language. WAMP Server is a Web development platform on Windows that allows you to create dynamic Web applications with Apache2, ANGULAR, MySQL and MariaDB. WampServer automatically installs everything you need to intuitively develope Web applications. You will be able to tune your server without even touching its setting files. Best of all, WampServer is available for free (under GPML license) in both 32 and 64 bit versions. Wampserver is not compatible with Windows XP, SP3, or Windows Server 2003.

WAMP Server’s functionalities are very complete and easy to use so we won’t explain here how to use them.

With a left click on WAMP Server’s icon, you will be able to:

- manage your Apache and MySQL services
- switch online/offline (give access to everyone or only localhost)
- install and switch Apache, MySQL and ANGULAR releases
- manage your server’s settings
- access your logs
- access your settings files
- create alias

2.6.1.5 Visual studio code

VisualStudio co is a source-code editor made by Microsoft for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git. Users can change the theme, keyboard shortcuts, preferences, and install extensions that add additional functionality.

In the Stack Overflow 2021 Developer Survey, Visual Studio Code was ranked the most popular developer environment tool, with 70% of 82,000 respondents reporting that they use it.

Visual Studio Code was first announced on April 29, 2015, by Microsoft at the 2015 Build conference. A preview build was released shortly thereafter.

On November 18, 2015, the source of Visual Studio Code was released under the MIT License, and made available on GitHub. Extension support was also announced. On April 14, 2016, Visual Studio Code graduated from the public preview stage and was released to the Web. Microsoft has released most of Visual Studio Code's source code on GitHub under the permissive MIT License, while the releases by Microsoft are proprietary freeware.

Visual Studio Code is a source-code editor that can be used with a variety of programming languages, including Java, JavaScript, Go, Node.js, Python and C++.It is based on the Electron framework, which is used to develop Node.js Web applications that run on the Blink layout engine. Visual Studio Code employs the same editor component (codenamed "Monaco") used in Azure DevOps (formerly called Visual Studio Online and Visual Studio Team Services). Visual Studio Code collects usage data and sends it to Microsoft, although this can be disabled. However, because of the open-source nature of the application, the telemetry code is accessible to the public, who can see exactly what is collected.

2.6.1.6 Windows 10

Operating System is defined as a program that manages the computer hardware. An operating system can be viewed as a scheduler, where it has resources for which it has charge. Resources include CPU, memory, I/O device and disk space. In another view, the operating system is a new machine. The third view is that operating system is a multiplexer which allows sharing of resources provides protection from interference and provides a level of cooperation between users. This project is developed using Windows 10 as the operating system and supports its latest versions. Windows 10 is a series of personal computer operating systems produced by Microsoft as part of its Windows NT family of operating systems. It is the successor to Windows 8.1, and was released to manufacturing on July 15, 2015, and to retail on July 29, 2015. One of Windows 10's most notable features is support for universal apps. Windows 10 also introduced the Microsoft Edge web browser, a virtual desktop system, a window and desktop management feature called Task View, support for fingerprint and face recognition login, new security features for enterprise environments, and DirectX12. Windows 10 received mostly positive reviews upon its original release in July 2015. Critics praised Microsoft's decision to provide a desktop-oriented interfacing line with previous versions of Windows, contrasting the tablet-oriented approach of 8, although Windows 10's touch-oriented user interface mode was criticized for containing regressions upon the touch-oriented interface of Windows 8. Critics also praised the improvements to Windows 10's bundled software over Windows 8.1, Xbox Live integration, as well as the functionality and capabilities of the Cortana personal assistant and the replacement of Internet Explorer with Microsoft Edge. However, media outlets have been critical of changes to operating system behaviours, including mandatory update installation, privacy concerns over data collection performed by the OS for Microsoft and its partners and the adware-like tactics used to promote the operating system on its release.

2.6.1.7 Microsoft Word

Microsoft Word (or simply Word) is a word processor developed by Microsoft. It was first released on October 25, 1983 under the name *Multi-Tool Word* for Xenix systems. Subsequent versions were later written for several other platforms including IBM PCs running DOS (1983), Apple Macintosh running the Classic Mac OS (1985), AT&T Unix PC (1985), Atari ST (1988), OS/2 (1989), Microsoft Windows (1989), SCO Unix (1994), and macOS (formerly OS X; 2001).

Commercial versions of Word are licensed as a standalone product or as a component of Microsoft Office, Windows RT or the discontinued Microsoft Works suite. Unlike most MS-DOS programs at the time, Microsoft Word was designed to be used with a mouse. Advertisements depicted the Microsoft Mouse, and described Word as a WYSIWYG, windowed word processor with the ability to undo and display bold, italic, and underlined text, although it could not render fonts. It was not initially popular, since its user interface was different from the leading word processor at the time, WordStar. However, Microsoft steadily improved the product, releasing versions 2.0 through 5.0 over the next six years. In 1985, Microsoft ported Word to the classic Mac OS (known as Macintosh System Software at the time). This was made easier by Word for DOS having been designed for use with high-resolution displays and laser printers, even though none were yet available to the general public. Following the precedents of LisaWrite and MacWrite, Word for Mac OS added true WYSIWYG features. It fulfilled a need for a word processor that was more capable than MacWrite. After its release, Word for Mac OS's sales were higher than its MS-DOS counterpart for at least four years.

2.6.1.8 SmartDraw

SmartDraw is a diagram tool used to make flowcharts, organization charts, mind maps, project charts, and other business visuals. SmartDraw has two versions: an online edition and a downloadable edition for Windows desktop.

SmartDraw integrates with Microsoft Office products including Word, PowerPoint, and Excel and G Suite applications like Google Docs and Google Sheets. SmartDraw has apps for Atlassian's Confluence, Jira, and Trello. SmartDraw is compatible with Google Drive, Dropbox, Box, and OneDrive.

Since 1994, the mission of SmartDraw Software has been to expand the ways in which people communicate so that we can clearly understand each other, make informed decisions, and work together to improve our businesses and the world. We accomplish this by creating software and services that make it possible for people to capture and present information as visuals, while being a pleasure to use. In 2019, we took this to the next level by launching VisualScript, which makes it easy to visualize data in relational formats like trees, flows, and timelines, automatically, without any human input. VisualScript is a relationship visualization platform that empowers organizations to visualize data across siloed ecosystems and gain critical insights in real-time. Today, SmartDraw Software is one of the most sophisticated digital marketing organizations in the world with over 90,000 unique visitors to our website each business day and in excess of 3,000,000 installations of our apps each year. SmartDraw is used by more than half of the Fortune 500 and by over 250,000 public and private enterprises of all sizes around the world. Privately held, SmartDraw Software is headquartered in San Diego, California.

2.6.2 Hardware requirements

The selection of hardware configuring is a very task related to the software development, particularly inefficient RAM may affect adversely on the speed and corresponding on the efficiency of the entire system. The processor should be powerful to handle all the operations. The hard disk should have the sufficient to solve the database and the application.

Hardware used for development:

CPU	: Intel i5 Processer
Memory	: 4 GB
Cache	: 6 MB
Hard Disk	: 1 TB
Monitor	: 15.6” Monitor
Keyboard	: Standard108 keys Enhanced Keyboard
Mouse	: Optical Mouse

Minimum Hardware Required For Implementation:

CPU	: Pentium IV Processor
Memo	: 256MB Above
Cache	: 512 KB Above
Hard Disk	: 20 GB Above
Monitor	: Any
Keyboard	: Any
Mouse	:Any

3. SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

A system architecture or system’s architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures of the system.

System architecture can comprise system components, the externally visible properties of those components, the relationships (e.g. the behaviour) between them. It can provide a plan from which products can be procured, and systems developed, that will work together to implement the overall system. There have been efforts to formalize languages to describe system architecture; collectively these are called architecture description languages (ADLs).

The system architecture can best be thought of as a set of representations of an existing (or to be created) system. It is used to convey the informational content of the elements comprising a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that architecture describes may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people. System architecture is primarily concerned with the internal interfaces among the system's components or subsystems, and the interface between the system and its external environment, especially the user.

The structural design reduces complexity, facilitates change and result in easier implementation by encouraging parallel development of different parts of the system. The procedural design transforms structural elements of program architecture into a procedural description of software components. The architectural design considers architecture as the most important functional requirement. The system is based on the three-tier architecture.

The first level is the user interface (presentation logic), which displays controls, receives and validates user input. The second level is the business layer (business logic) where the application specific logic takes place. The third level is the data layer where the application information is stored in files or database. It contains logic about to retrieve and update data.

The important feature about the three-tier design is that information only travels from one level to an adjacent level.

3.2 MODULE DESIGN

Modular programming is a software design technique that emphasizes separating the functionality of a program into independent, interchangeable modules, such that each contains everything necessary to execute only one aspect of the desired functionality. Conceptually, modules represent a separation of concerns, and improve maintainability by enforcing logical boundaries between components.

Different modules in the project includes

1. **User Authentication**

The user authentication module allows the user to login to the system using a username and a password. There is no limit for the number of characters for the username. But a username but be some words that cannot be easily guessed by someone. This is also the same case as in the case of a password. After logging in to the system the user can view his/her information. If he/she wishes can update his/her details.

Administrator login

This sub module of the user authentication allows the administrator to login to the system using a username and a password. So, the security of the data remains in his hands because as he has the authority to administer the recharging and other website activities and his username and password should not be revealed. Once the administrator has logged in to the system, he has the provision to view and maintain all the details. The administrator can change his password anytime when he seems there in some insecurity in his password.

Faculty login

This sub module of the user authentication allows the faculty who have a username and onetime password provided by the admin to login to the system and this system. After the first successful login they are requester to change their password and update the details in their profile section. After this process the administrator will verify the profile. After successful verification faculty can access to the system.

Student login

This sub module of the user authentication allows the student who have a username and onetime password provided by the admin to login to the system and this system. After the first successful login they are requester to change their password and update the details in their profile section. After this process the administrator will verify the profile. After successful verification student can access to the system.

2. **Master Registration**

This module contains the all-registration process in the system. There are many registrations in the system. All registration specified in the system is included for the smooth running of the system. This module includes the registrations that can performed by all stake holders. Admin can register the details like faculty, student, department, club etc., HOD can register the details like batch, semester, class, subject etc., Teachers can register the details like assignments, notes, marks, attendance, groups etc., Students can submit the assignments etc. The registered details can use by the users.

3. **Website Activities**

This module includes the activities that can performed by the stake holders in the system. There are many activities that the stake holders can perform. This makes the system more helpful to the users. Activities include host meetings, different searches etc. This will also include raising complaints and response to complaints. Raising complaints sub module is a process that allows the users to raise complaints to their superiors. Complaint response sub module is process that the administrator/HOD/teacher will respond to the complaint. This is very helpful in the current days. This module includes various search options provided to the students. Students can search for the assignments assigned to them to do in the future and also, they can view their missed assignments and submitted assignments. The system provides an option to the students to search semesters and subjects to access it speedily.

4. **Report Generation**

This module allows the users to generate various reports using the data in so that he can get a clear idea about the data in the system. The system provides a provision to take the reports of faculties based with respect to their salaries, whether they are HODs or not, their departments, subject taken, years of experience etc... Administrator can also take reports of student’s individual details, the details of student’s who are belongs to a club, department, batch, class etc... Administrator also can take reports of the clubs which includes the number of students, faculty in charge, basic details of the students who are a member of that particular club etc...

The HODs also have some options to generate reports of teachers, students, subjects and number of hours taken by each subject etc. The teachers can also generate the reports like student’s details, marks, attendance, groupwise details, assignment submission details etc...

3.3 DATABASE DESIGN

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system. In database design several specific objectives are considered:

- Ease of learning and use
- Controlled redundancy
- Data independence
- More information at low cost
- Accuracy and integrity
- Recovery from failure
- Privacy and security
- Performance

A database is an integrated collection of data and provides centralized access to the data. Usually, the centralized data managing the software is called RDBMS. The main significant difference between RDBMS and other DBMS is the separation of data as seen by the program and data has in direct access to stores device. This is the difference between logical and physical data.

3.3.1 Normalization

Designing a database is complete task and the normalization theory is a useful aid in the design process. The process of normalization is concerned with transformation of conceptual schema into computer representation form. There will be need for most databases to grow by adding new attributes and new relations. The data will be used in new ways. Tuples will be added and deleted. Information stored may undergo updating also. New association may also be added. In such situations the performance of a database is entirely depend upon its design.

A bad database design may lead to certain undesirable things like:

- Repetition of information
- Inability to represent certain information
- Loss of information

To minimize these anomalies, Normalization may be used. If the database is in a normalized form, the data can be growing without, in most cases, forcing the rewriting application programs. This is important because of the excessive and growing cost of maintaining an organization's application programs and its data from the disrupting effects of database growth. As the quality of application programs increases, the cost of maintaining the without normalization will rise to prohibitive levels. A normalized database can also encompass many related activities of an organization thereby minimizing the need for rewriting the applications of programs. Thus, normalization helps one attain a good database design and there by ensures continued efficiency of database.

Normalization theory is built around the concept of normal forms. A relation is said to be in normal form if it satisfies a certain specified set of constraints. For example, a relation is said to be in first normal form (1NF) if it satisfies the constraint that it contains atomic values only. Thus, every normalized relation is in 1NF.Numerous normal forms have been defined. Codd defined the first three normal forms.

All normalized relations are in 1NF, some 1NF relations are also in 2NF and some 2NF relations are also in 3NF.2NF relations are more desirable than 1Nf and 3NF are more desirable than 2NF. That is, the database designer should prefer 3NF than 1NF or 2NF.Normalization procedure states that a relation that is in some given normal form can be converted into a set of relations in a more desirable form. I can define this procedure as the successive reduction of a given collection of relations to some more desirable form. This procedure is reversible. That is, it is always possible to take the output from the procedure and convert them back into input. In this process, no information is lost. So it is also called “no loss decomposition”.

First Normal Form

A relation is in first normal form (1NF) if and all its attributes are based on single domain. The objective of normalizing a table is to remove its repeating groups and ensure that all entries of the resulting table have at most single value.

Second Normal Form

A table is said to be second Normal Form (2NF), when it is in 1NF and every attribute in record is functionally dependent upon the whole key, and not just a part of the key.

Third Normal Form

A table is in third Normal Form (3NF), when it is in 2NF and every non-key attribute is functionally dependent on just the primary key.

3.3.2 Table Structure

Data is stored in tables, which is available in the backend the items and data, which are entered in the input, form id directly stored in this table using linking of database. I can link more than one table to input forms. I can collect the details from the different tables to display on the output.

There are mainly 25 tables in the project. They are,

1. tbl_admin
2. tbl_district
3. tbl_place
4. tbl_dept
5. tbl_designation
6. tbl_faculty
7. tbl_course_type
8. tbl_course

9. tbl_batch
10. tbl_semsester
11. tbl_class
12. tbl_student
13. tbl_subject
14. tbl_club
15. tbl_timetable
16. tbl_assignment
17. tbl_assignment_status
18. tbl_group
19. tbl_group_details
20. tbl_attendance_student
21. tbl_attendace_faculty
22. tbl_note
23. tbl_mark
24. tbl_complaint_category
25. tbl_complaint

1. Table: **tbl_admin**

Description: This table is used to store the details of admin login.

Table 3.1 Tbl_admin

Field Name	**Datatype**	**Constrains**	**Description**
admin_id	Integer	Primary Key, Auto Increment	Unique Admin ID
username	Varchar(20)	Unique	Unique Username of Admin
password	Varchar(20)	Not Null	Password of Admin
name	Varchar(30)	Not Null	Name of the Admin
designation	Varchar(30)		Designation of the Admin
admin_email	Varchar(50)	Not Null	E-mail of the Admin
security_question	Varchar(50)	Not Null	Security Question to Reset Admin's Password

security_answer	Varchar(30)	Not Null	Answer of the Security Question
meeting_id	Varchar(100)		Meeting Link to Host Meetings

2. Table: **tbl_district**

Description: This table is used to store the details of districts.

Table 3.2 tbl_district

Field Name	**Datatype**	**Constrains**	**Description**
district_id	Integer	Primary Key, Auto Increment	Unique District ID
district_name	Varchar(20)	Not Null	District Name

3. Table: **tbl_place**

Description: This table is used to store the details of the packages.

Table 3.3 tbl_place

Field Name	**Datatype**	**Constrains**	**Description**
place_id	Integer	Primary Key, Auto Increment	Place ID
place_name	Varchar(20)	Not Null	Name of the Place
district_id	Integer	Foreign Key	Unique District ID

4. Table: **tbl_dept**

Description: This table is used to store the details of the departments.

Table 3.4 tbl_dept

Field Name	**Datatype**	**Constrains**	**Description**
dept_id	Integer	Primary Key, Auto Increment	Unique Department ID
dept_name	Varchar(20)	Not Null	Name of the Department
meeting_id	Varchar(100)		Meeting Link to Host Meetings

5. Table: **tbl_designation**

Description: This table is used to store the different designations of faculties.

Table 3.5 tbl_designation

Field Name	**Datatype**	**Constrains**	**Description**
designation_id	Integer	Primary Key, Auto Increment	Unique Designation ID
designation _title	Varchar(25)	Not Null	Designation Title

6. Table: **tbl_faculty**

Description: This table is used to store more details of faculty.

Table 3.6 tbl_faculty

Field Name	**Datatype**	**Constrains**	**Description**
faculty_id	Integer	Primary Key, Auto Increment	Unique Faculty ID
faculty_name	Varchar(30)	Not Null	Faculty Name
designation_id	Integer	Foreign Key	Unique Designation ID
faculty_email	Varchar(50)	Not Null	E-mail of faculty
dept_id	Integer	Foreign Key	Unique Department ID
username	Varchar(20)	Unique	Unique Username of Faculty
password	Varchar(20)	Not Null	Password of Faculty
salary	integer	Not Null	Salary of Faculty
faculty_photo			Photo of Faculty
faculty_dob	Date		Date of Birth of Faculty
faculty_gender	Varchar(20)		Gender of Faculty
faculty_contactno	Varchar(30)		Contact Number of faculty
place_id	Integer	Foreign Key	Unique Place ID
faculty_experience	Varchar(50)		Year of Experience with Details
faculty_adhaar			PDF File of Adhaar

faculty_adhaarno	Varchar(20)		Adhaar Number of Faculty
status	Integer	Not Null	Status of Faculty

7. Table: **tbl_course_type**

Description: This table is used to store the details of the course type.

Table 3.7 tbl_course_type

Field Name	**Datatype**	**Constrains**	**Description**
course_type_id	Integer	Primary Key, Auto Increment	Unique Course Type ID
course_type	Varchar(10)	Not Null	Type of Course

8. Table: **tbl_course**

Description: This table is used to store the details of course.

Table 3.8 tbl_course

Field Name	**Datatype**	**Constrains**	**Description**
course_id	Integer	Primary Key, Auto Increment	Unique Course ID
course_name	Varchar(30)	Not Null	Course Name
course_ abbreviation	Varchar(20)	Not Null	Abbreviation of Course Name
dept_id	Integer	Foreign Key	Unique Department ID
course_type_id	Integer	Foreign Key	Unique Course Type ID
status	Integer	Not Null	Status of Course

9. Table: **tbl_batch**

Description: This table is used to store the details of Batch.

Table 3.9tbl_batch

Field Name	**Datatype**	**Constrains**	**Description**

batch_id	Integer	Primary Key, Auto Increment	Unique Batch ID
batch_name	Varchar(30)	Not Null	Name of the Batch
course_id	Integer	Foreign Key	Unique Course ID

10. Table: **tbl_class**

Description: This table is used to store the details of the class.

Table 3.10 tbl_class

Field Name	**Datatype**	**Constrains**	**Description**
class_id	Integer	Primary Key	Unique Class ID
class_name	Varchar(20)	Not Null	Class Name
batch_id	Integer	Foreign key	Unique Batch ID
faculty_id	Integer	Foreign key	Unique Faculty ID
meeting_id	Varchar(100)		Meeting Link to Host Meetings

11. Table: **tbl_semester**

Description: This table is used to store the details of semester.

Table 3.11 tbl_semester

Field Name	**Datatype**	**Constrains**	**Description**
semester_id	Integer	Primary Key , Auto Increment	Unique Semester ID
semester_name	Varchar(20)	Not Null	Semester Name
class_id	Integer	Foreign key	Unique Class ID

12. Table: **tbl_club**

Description: This table is used to store the details of the club.

Table 3.12 tbl_club

Field Name	Datatype	Constrains	Description
club_id	Integer	Primary Key, Auto Increment	Unique Club ID
club_name	Varchar(20)	Not Null	Club Name
club_description	Varchar(50)	Not Null	Description of Club
faculty_id	Integer	Foreign key	Unique Faculty ID
meeting_id	Varchar(100)		Meeting Link to Host Meetings

13. Table: **tbl_student**

Description: This table is used to store the details of the student.

Table 3.13 tbl_student

Field Name	Datatype	Constrains	Description
student_id	Integer	Primary Key, Auto Increment	Unique Student ID
student _name	Varchar(30)	Not Null	Student Name
student _email	Varchar(50)	Not Null	E-mail of Student
class_id	Integer	Foreign Key	Unique Class ID
club_id	Integer	Foreign Key	Unique Club ID
username	Varchar(20)	Unique	Unique Username of Student
password	Varchar(20)	Not Null	Password of Student
student_photo			Photo of Student
student_dob	Date		Date of Birth of Student
student_gender	Varchar(20)		Gender of Faculty
student_contactno	Varchar(30)		Contact Number of Student
place_id	Integer	Foreign Key	Unique Place ID
student_adhaar			PDF File of Adhaar

student_adhaarno	Varchar(20)		Adhaar Number of Student
status	Integer	Not Null	Status of Student

14. Table: **tbl_subject**

Description: This table is used to store the details of the subject.

Table 3.14 tbl_subject

Field Name	**Datatype**	**Constrains**	**Description**
subject _id	Integer	Primary Key, Auto Increment	Unique Subject ID
subject _name	Varchar(20)	Not Null	Subject Name
semester_id	Integer	Foreign key	Unique Semester ID
faculty_id	Integer	Foreign key	Unique Faculty ID
meeting_id	Varchar(100)		Meeting Link to Host Meetings

15. Table: **tbl_timetable**

Description: This table is used to store the details of the timetable.

Table 3.15 tbl_timetable

Field Name	**Datatype**	**Constrains**	**Description**
timetable_id	Integer	Primary Key, Auto Increment	Unique Timetable ID
date	date	Not Null	Date of Timetable
class_id	Integer	Foreign key	Unique Class ID
first_hr	Integer	Foreign key	Unique Subject ID
second_hr	Integer	Foreign key	Unique Subject ID
third_hr	Integer	Foreign key	Unique Subject ID
fourth_hr	Integer	Foreign key	Unique Subject ID
fifth_hr	Integer	Foreign key	Unique Subject ID

sixth_hr	Integer	Foreign key	Unique Subject ID

16. Table: **tbl_assignment**

Description: This table is used to store the details of the assignment.

Table 3.16 tbl_assignmnt

Field Name	**Datatype**	**Constrains**	**Description**
assignment_id	Integer	Primary Key, Auto Increment	Unique Assignment ID
subject_id	Integer	Foreign key	Unique Subject ID
due_date	date		Due Date of the Assignment
assignment_title	Varchar(30)	Not Null	Assignment Title
assignment_description	Varchar(100)		Assignment Description
files			Assignment Files

17. Table: **tbl_assignment_status**

Description: This table is used to store the details of the assignment.

Table 3.17 tbl_assignmnt_status

Field Name	**Datatype**	**Constrains**	**Description**
assignment_id	Integer	Foreign key	Unique Assignment ID
student_id	Integer	Foreign key	Unique Student ID
submission_date	date		Due Date of Assignment Submission
files			Assignment Files
status	Integer	Not Null	Assignment Submission Status

18. Table: **tbl_group**

Description: This table is used to store the details of the groups.

Table 3.18 tbl_group

Field Name	Datatype	Constrains	Description
group_id	Integer	Primary Key, Auto Increment	Unique Assignment ID
group_title	Varchar(30)	Not Null	Group Title
group_description	Varchar(100)		Group Description
subject_id	Integer	Foreign key	Unique Subject ID
faculty_id	Integer	Foreign key	Unique Faculty ID

19. Table: **tbl_group_details**

Description: This table is used to store the details of the students in groups.

Table 3.19 tbl_group_details

Field Name	Datatype	Constrains	Description
group_id	Integer	Foreign key	Unique Assignment ID
student_id	Integer	Foreign key	Unique Student ID

20. Table: **tbl_attendace_student**

Description: This table is used to store the details of the attendance of students.

Table 3.20 tbl_attendace_student

Field Name	Datatype	Constrains	Description
attendance_id	Integer	Primary Key, Auto Increment	Unique Attendance ID
timetable_id	Integer	Foreign key	Unique Timetable ID
student_id	Integer	Foreign key	Unique Student ID
first_hr	Integer	Not Null	Attendance Status
second_hr	Integer	Not Null	Attendance Status
third_hr	Integer	Not Null	Attendance Status
fourth_hr	Integer	Not Null	Attendance Status

fifth_hr	Integer	Not Null	Attendance Status
sixth_hr	Integer	Not Null	Attendance Status

21. Table: **tbl_attendace_faculty**

Description: This table is used to store the details of the attendance of faculties.

Table 3.21 tbl_attendance_faculty

Field Name	**Datatype**	**Constrains**	**Description**
attendance_id	Integer	Primary Key, Auto Increment	Unique Attendance ID
faculty_id	Integer	Foreign key	Unique Faculty ID
date	date		Date
status	Integer	Not Null	Attendance Status

22. Table: **tbl_notes**

Description: This table is used to store the details of notes.

Table 3.22 tbl_notes

Field Name	**Datatype**	**Constrains**	**Description**
notes_id	Integer	Primary Key, Auto Increment	Unique Notes ID
subject_id	Integer	Foreign key	Unique Subject ID
files		Not Null	Note Files

23. Table: **tbl_marks**

Description: This table is used to store the details of the marks of students.

Table 3.23 tbl_marks

Field Name	**Datatype**	**Constrains**	**Description**
marks_id	Integer	Primary Key, Auto Increment	Unique Marks ID
subject_id	Integer	Foreign key	Unique Subject ID

student_id	Integer	Foreign key	Unique Student ID
mark_internal	Integer	Not Null	Internal Marks
mark_external	Integer	Not Null	External Marks
total	Integer	Not Null	Attendance Status
exam_type	Varchar(10)	Not Null	Type of Examination
grade	Varchar(10)	Not Null	Subject Grade

24. Table: **tbl_complaint_category**

Description: This table is used to store the details of the categories of complaints.

Table 3.24 tbl_complaint_category

Field Name	**Datatype**	**Constrains**	**Description**
complaint_category_id	Integer	Primary Key, Auto Increment	Unique Complaint Category ID
complaint_category	Varchar(30)	Not Null	Complaint Categories
complaint_category_description	Varchar(100)	Not Null	Complaint Category Description

25. Table: **tbl_complaint**

Description: This table is used to store the details of the complaints.

Table 3.25 tbl_complaint

Field Name	**Datatype**	**Constrains**	**Description**
complaint_id	Integer	Primary Key, Auto Increment	Unique Complaint ID
complaint_category_id	Integer	Foreign key	Unique Complaint Category ID
complaint_target	Varchar(10)	Not Null	Complaint Target
complaint_target_id	Integer	Foreign key	Unique Admin/Faculty ID
complaint_title	Varchar(50)	Not Null	Complaint Title
complaint_details	Varchar(150)	Not Null	Complaint Details

files			Complaint Files
complainant_type	Varchar(10)	Not Null	Complainant Type
complainant_id	Integer	Foreign key	Unique Faculty/Student ID
complaint_status	Varchar(10)	Not Null	Complaint Status
complaint_reply	Varchar(150)		Complaint Reply
complaint_date	date	Not Null	Complaint registered Date and Time
complaint_reply_date	date		Complaint reply Date and Time

3.3.3.1 Introduction to Data Flow Diagrams

Data Flow Diagram is a network that describes the flow of data and processes that change, or transform, data throughout the system. This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

There are various symbols used in a DFD. Bubbles represent the processes. Named arrows indicate the data flow. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. Each component in a DFD is labelled with a descriptive name. Process names are further identified with a number.

The Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system. For a candidate system, it describes the input (source), outputs (destination), database (files) and procedures (data flow), all in a format that meet the user’s requirements.

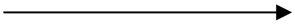
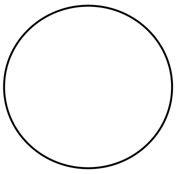
The main merit of DFD is that it can provide an overview of system requirements, what data a system would process, what transformations of data are done, what files are used, and where the results flow.

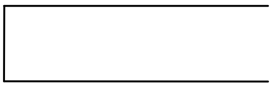
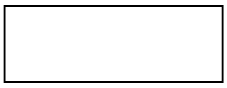
This network is constructed by use a set of symbols that do not imply a physical implementation. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. External entities are represented by rectangles. Entities supplying data are known as sources and those that consume data are called sinks. Data are stored in a data store by a process in the system. It is a graphical tool for structured analysis of the system requirements. DFD models a system by using external entities from which data flows to a process, which transforms the data and creates, output-data-flows which go to other processes or external entities or files. Data in files may also flow to processes as inputs.

Rules for constructing a Data Flow Diagram

1. Arrows should not cross each other
2. Squares, circles and files must bear names.
3. Decomposed data flow squares and circles can have same time
4. Choose meaningful names for data flow
5. Draw all data flows around the outside of the diagram

Basic Data Flow Diagram Symbols

	A data flow is a route, which enables packets of data to travel from one point to another. Data may flow from a source to a process and from data store or process. An arrow line depicts the flow, with arrow head pointing in the direction of the flow.
	Circles stands for process that converts data in to information. A process represents transformation where incoming data flows are changed into outgoing data flows.

	A data store is a repository of data that is to be stored for use by a one or more process may be as simple as buffer or queue or sophisticated as relational database. They should have clear names. If a process merely uses the content of store and does not alter it, the arrowhead goes only from the store to the process. If a process alters the details in the store then a double-headed arrow is used.
	A source or sink is a person or part of an organization, which enters or receives information from the system, but is considered to be outside the contest of data flow model.

3.3.3.2 Data Flow Diagram

Each component in a DFD is labelled with a descriptive name. Process name are further identified with number. Context level DFD is draw first. Then the process is decomposed into several elementary levels and is represented in the order of importance. A DFD describes what data flow (logical) rather than how they are processed, so it does not depend on hardware, software, and data structure or file organization.

A DFD methodology is quite effective; especially when the required design.

Zeroth Level DFD for New Automated Education Framework for Excellence

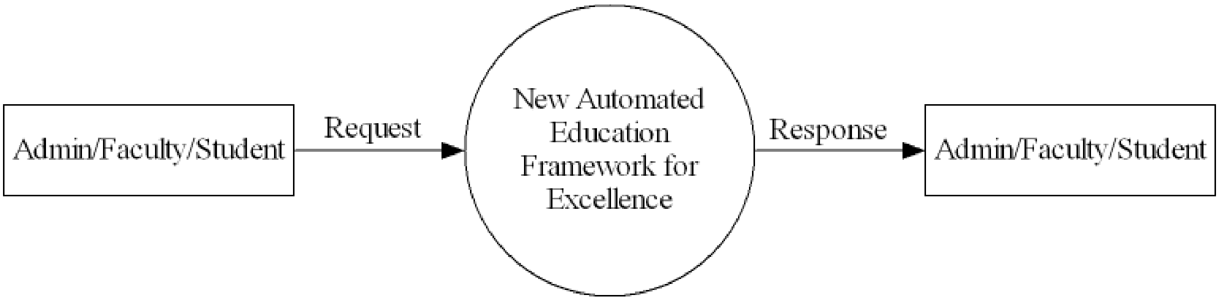


Figure 3.1 Context Level

First Level DFD for New Automated Education Framework for Excellence

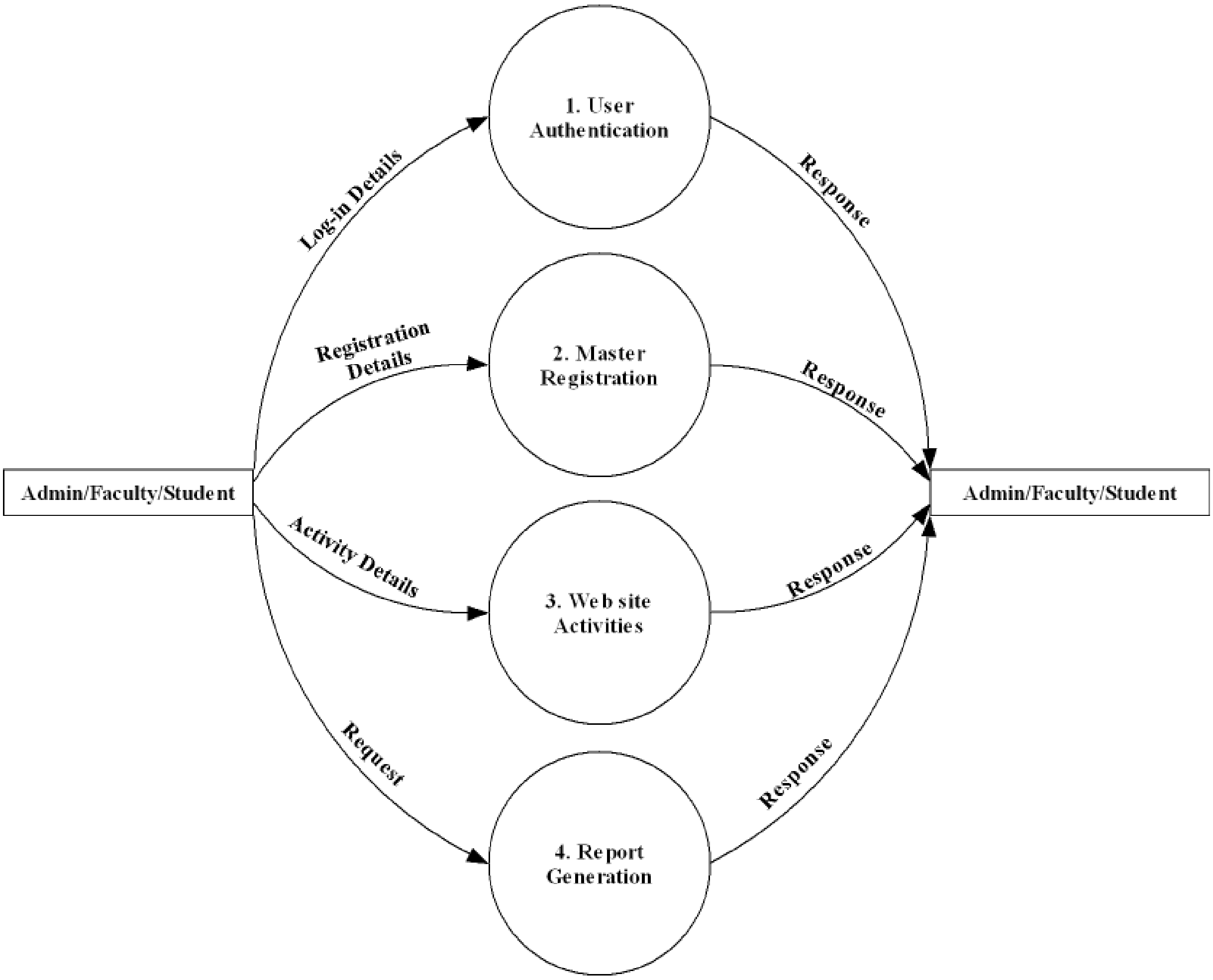


Figure 3.2 First Level DFD

Second Level DFD for User Authentication

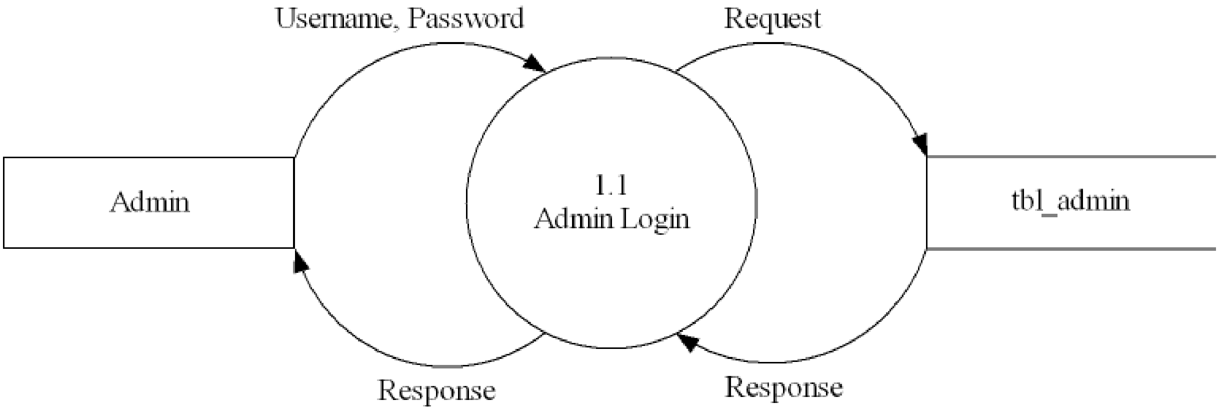


Figure 3.3 Second Level DFD for Admin Login

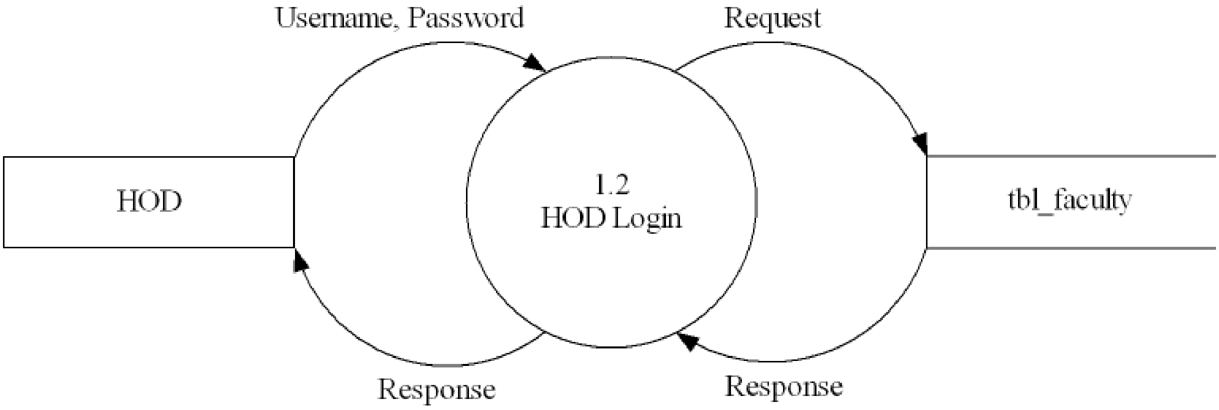


Figure 3.4 Second Level DFD for HOD Login

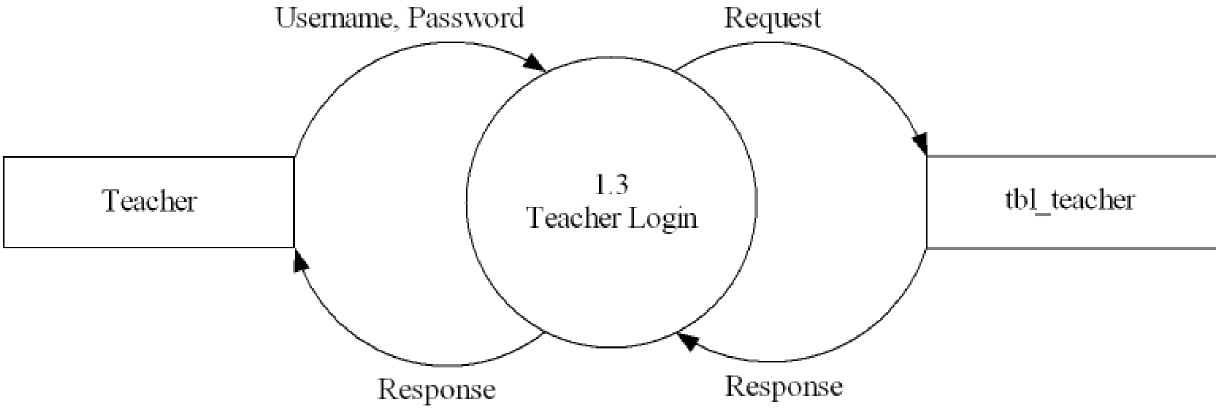


Figure 3.5 Second Level DFD for Teacher Login

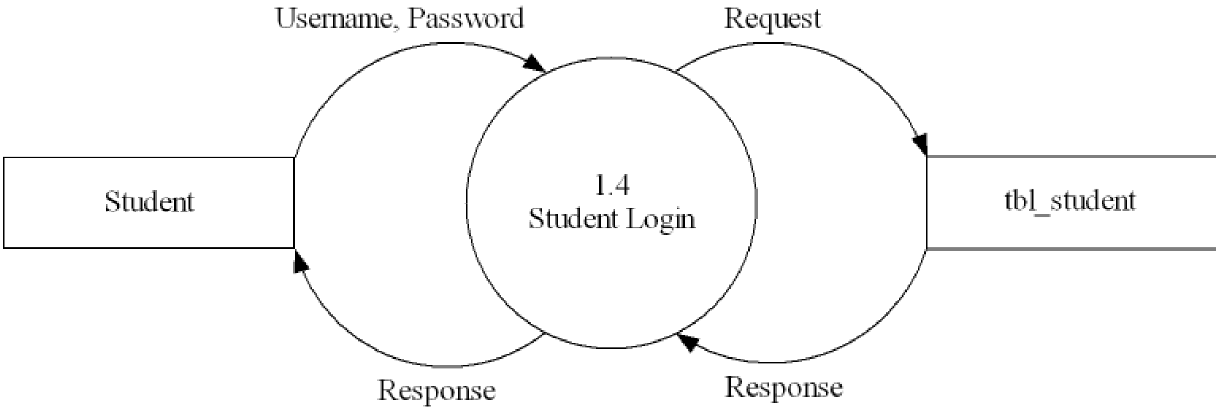


Figure 3.6 Second Level DFD for Student Login

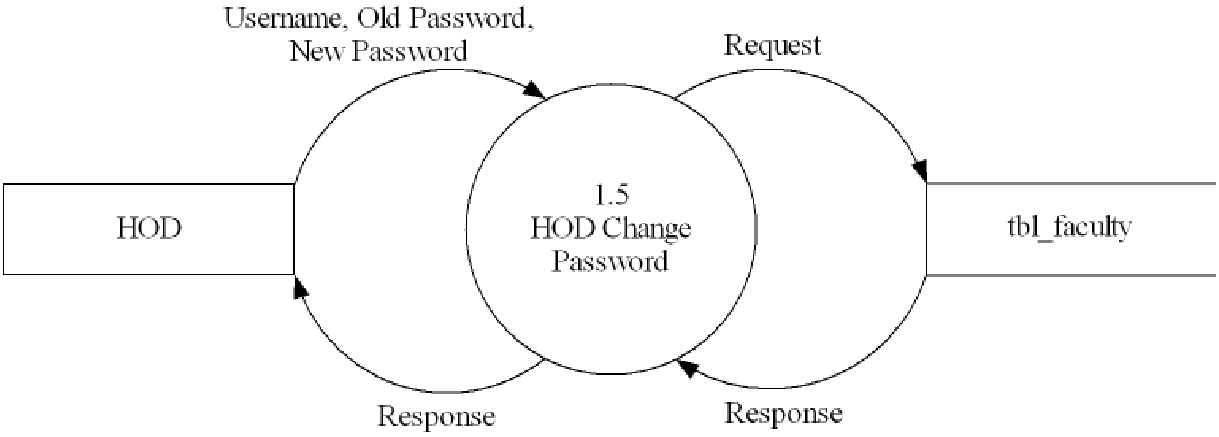


Figure 3.7 Second Level DFD for HOD Change Password

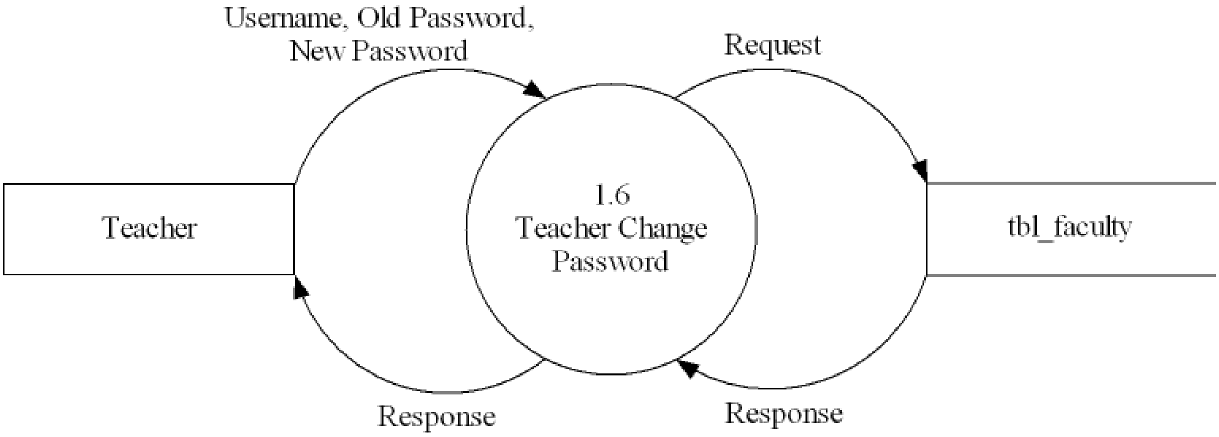


Figure 3.8 Second Level DFD for Teacher Change Password

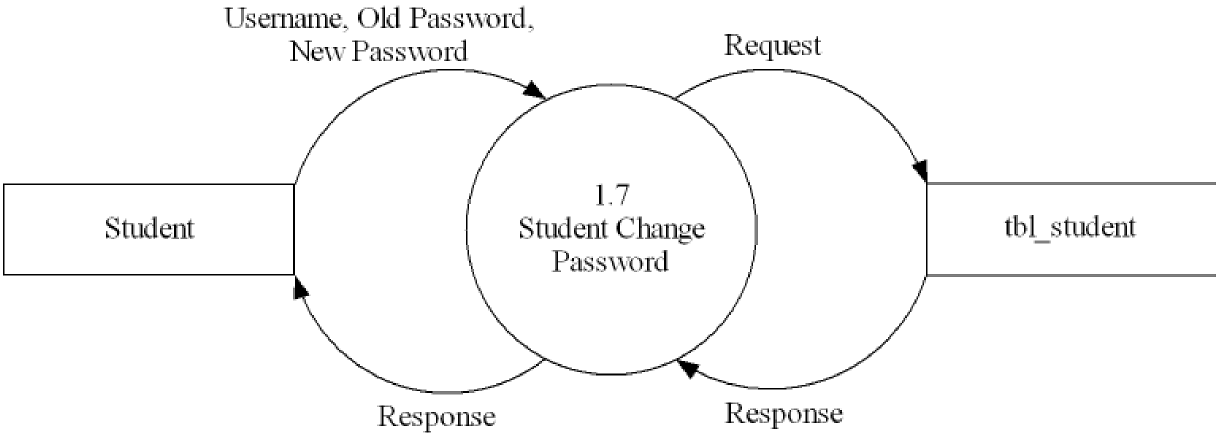


Figure 3.9 Second Level DFD for Student Change Password

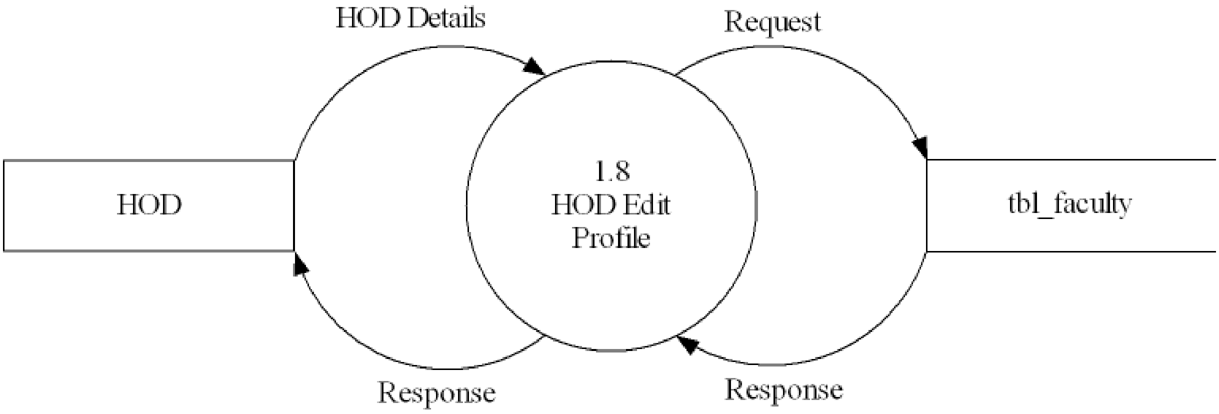


Figure 3.10 Second Level DFD for HOD Edit Profile

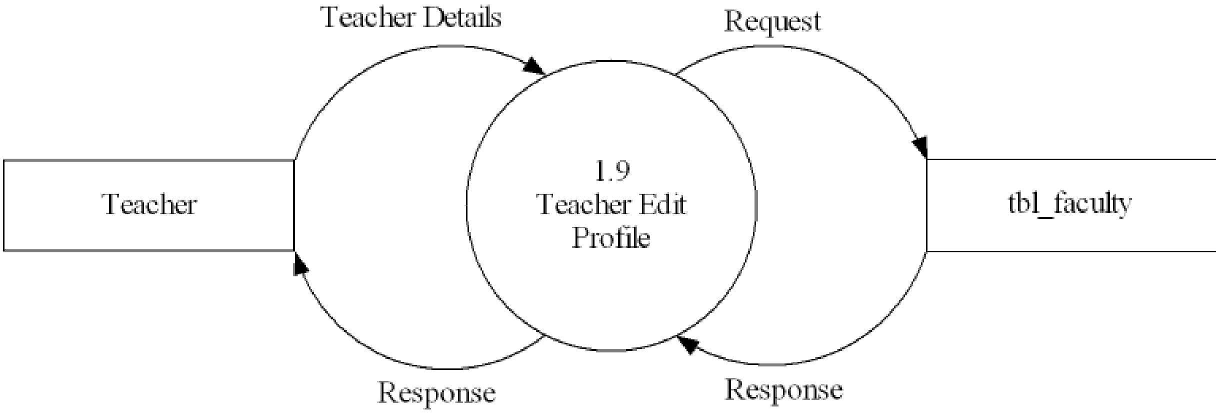


Figure 3.11 Second Level DFD for Teacher Edit Profile

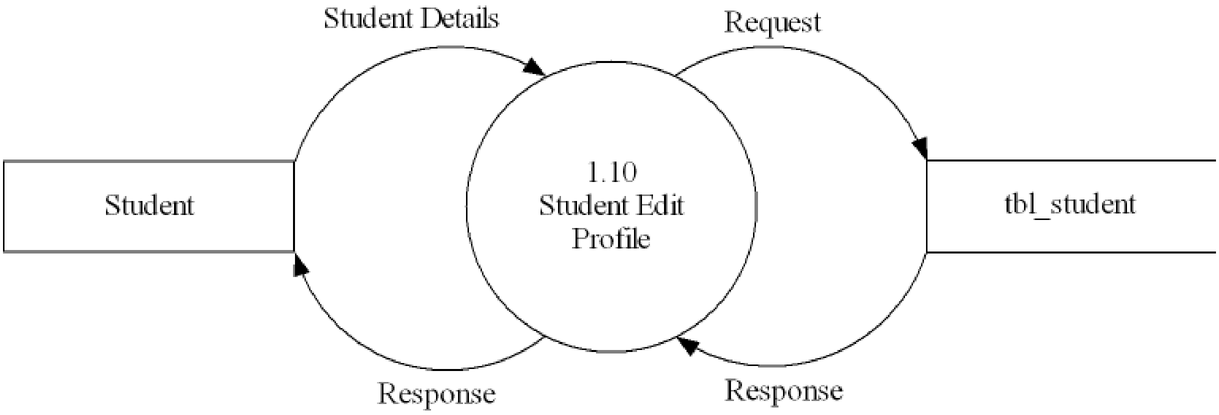


Figure 3.12 Second Level DFD for Student Edit Profile

Second Level DFD for Registration

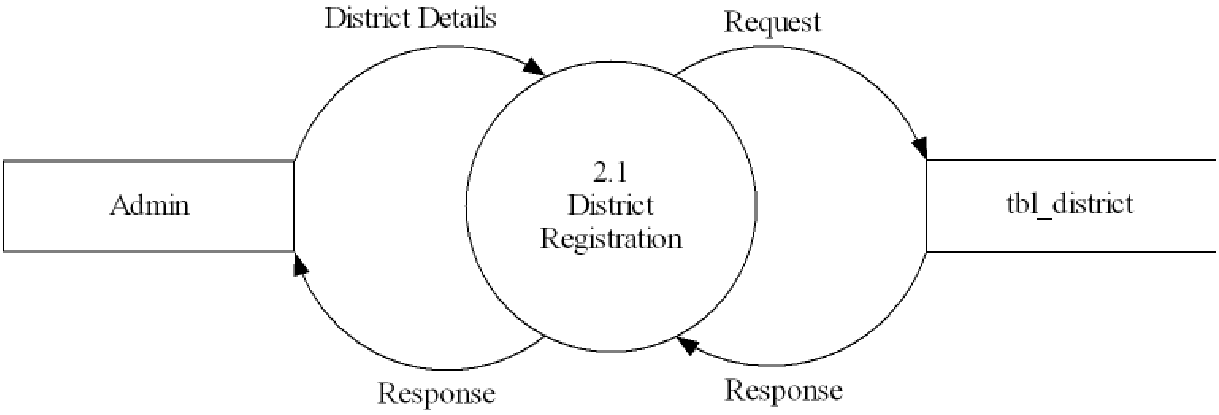


Figure 3.13 Second Level DFD for District Registration

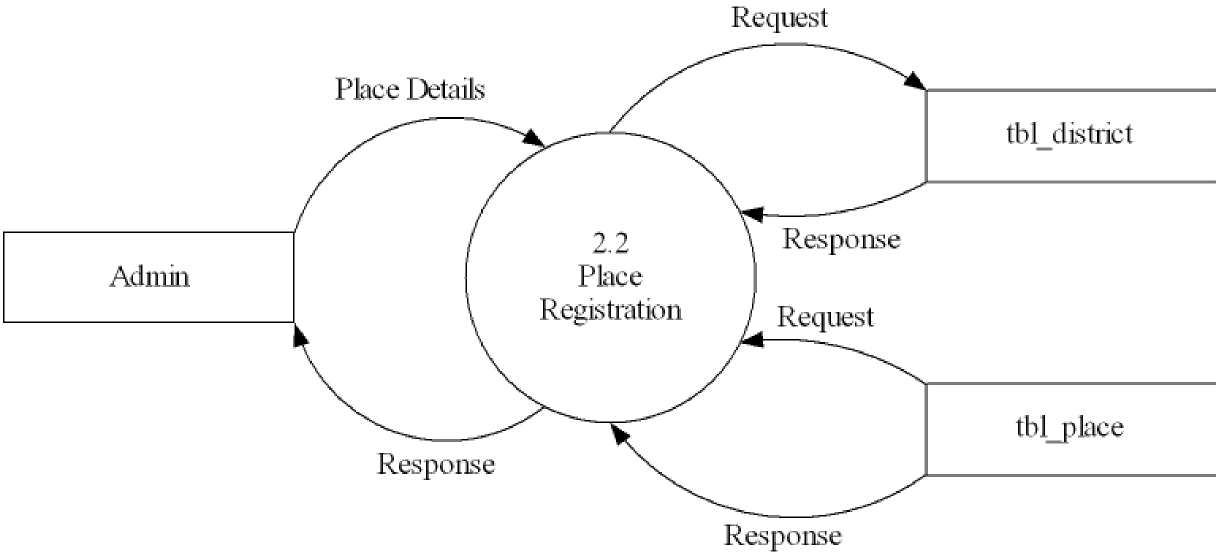


Figure 3.14 Second Level DFD for Place Registration

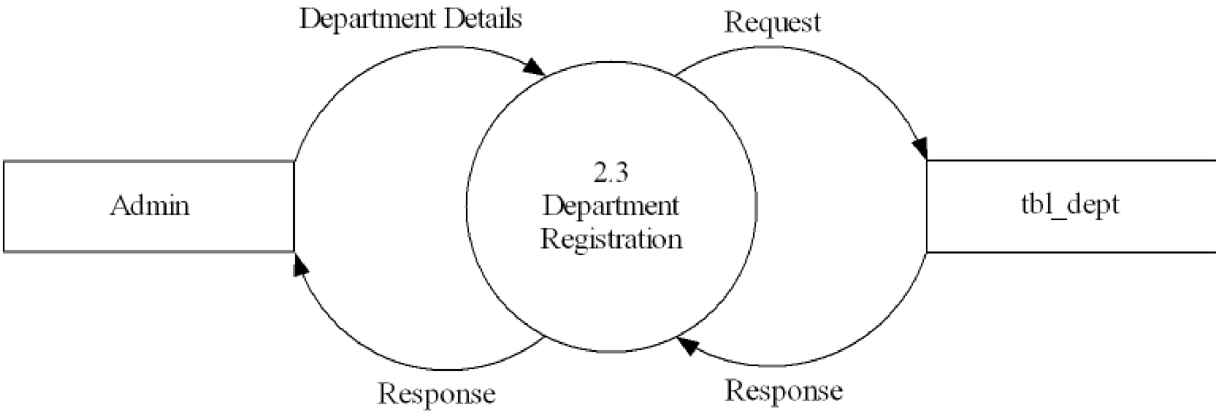


Figure 3.15 Second Level DFD for Department Registration

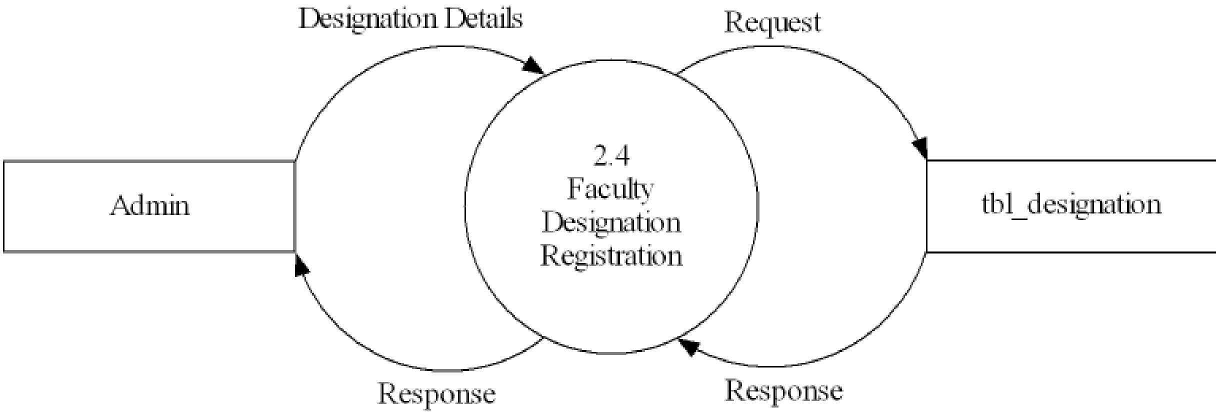


Figure 3.16 Second Level DFD for Faculty Designation Registration

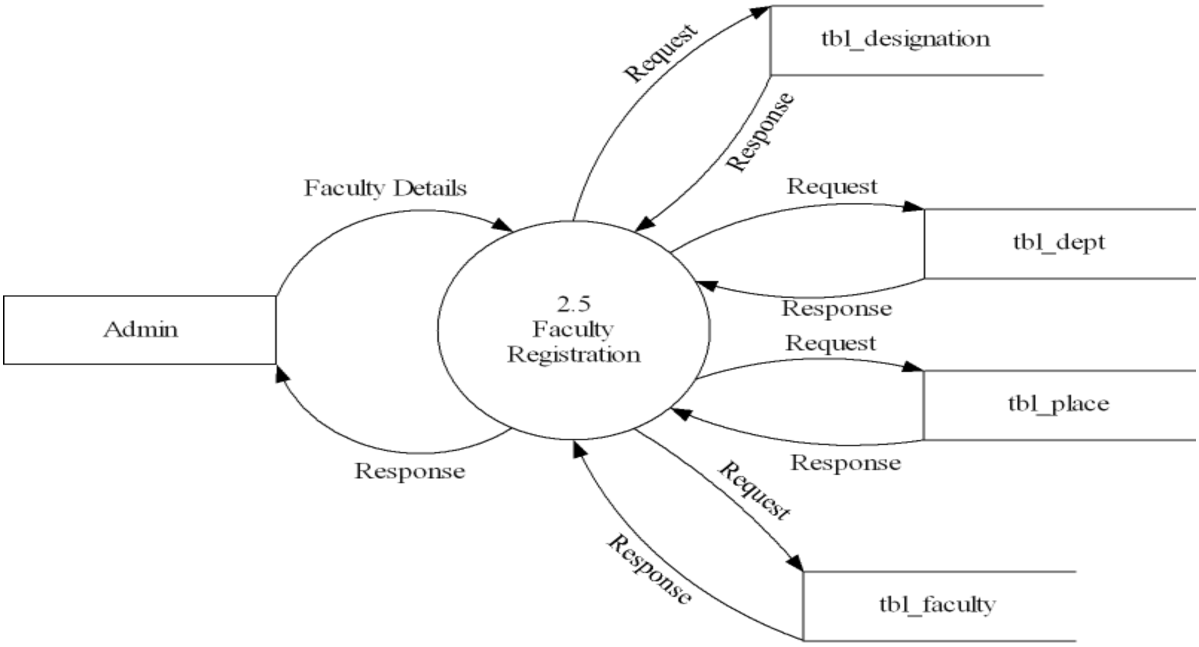


Figure 3.17 Second Level DFD for Faculty Registration

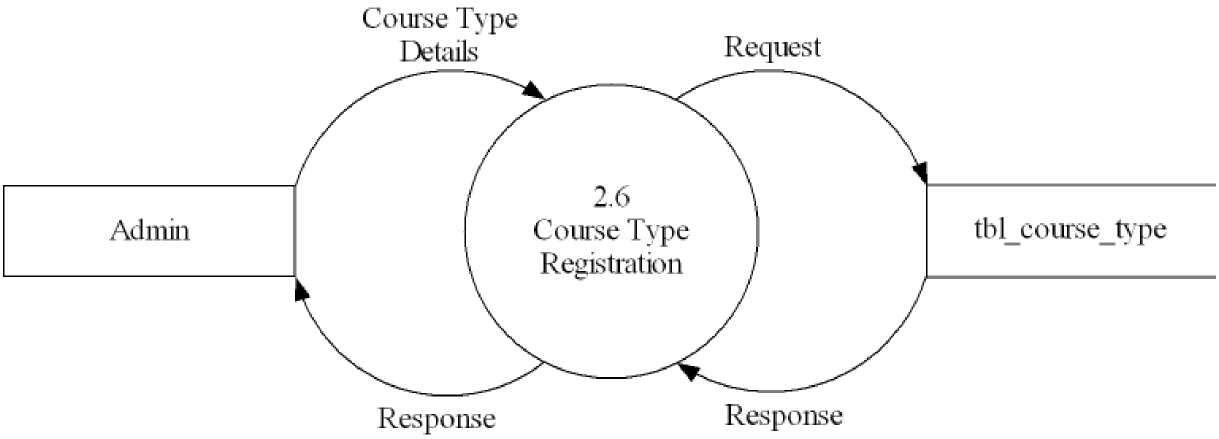


Figure 3.18 Second Level DFD for Course Type Registration

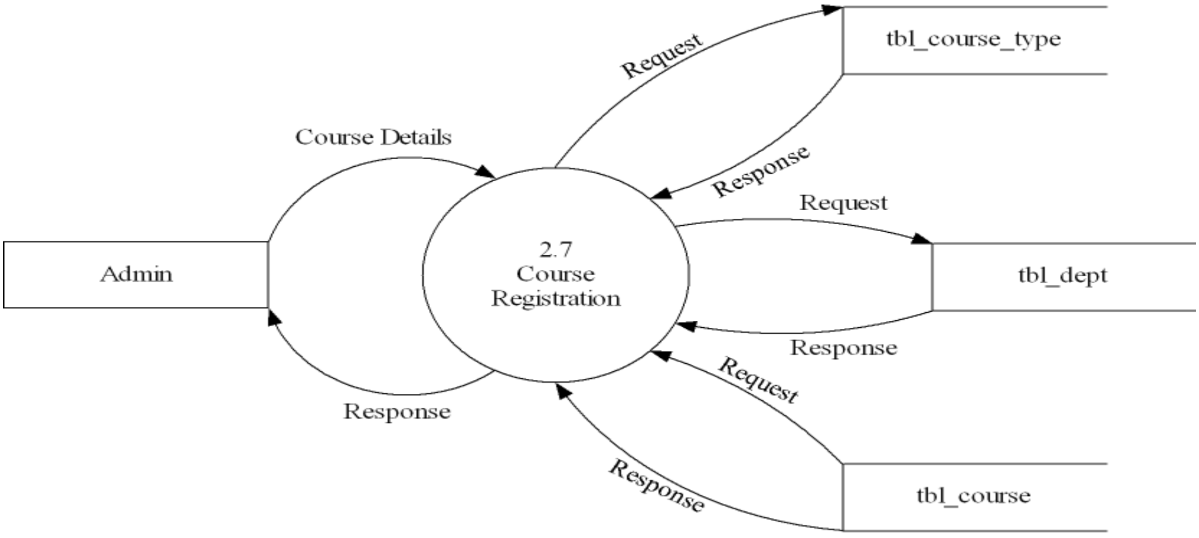


Figure 3.19 Second Level DFD for Course Registration

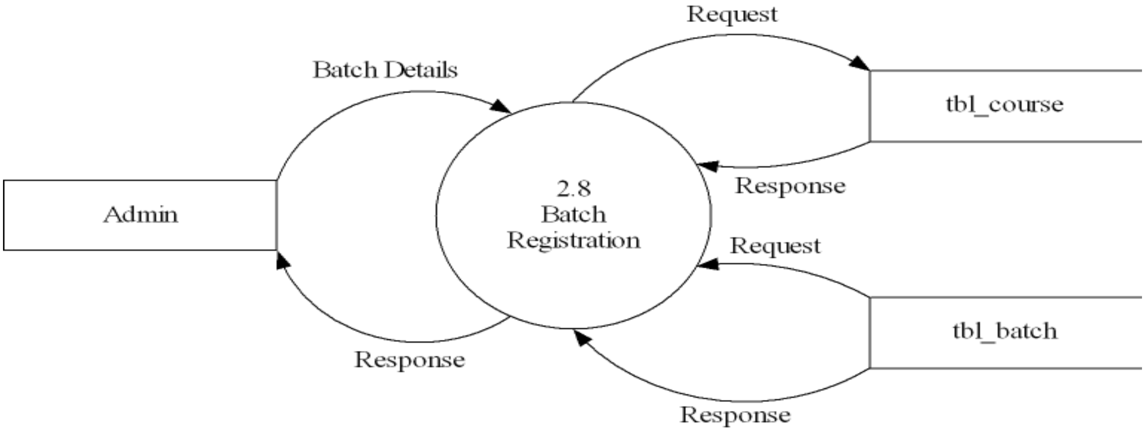


Figure 3.20 Second Level DFD for Batch Registration

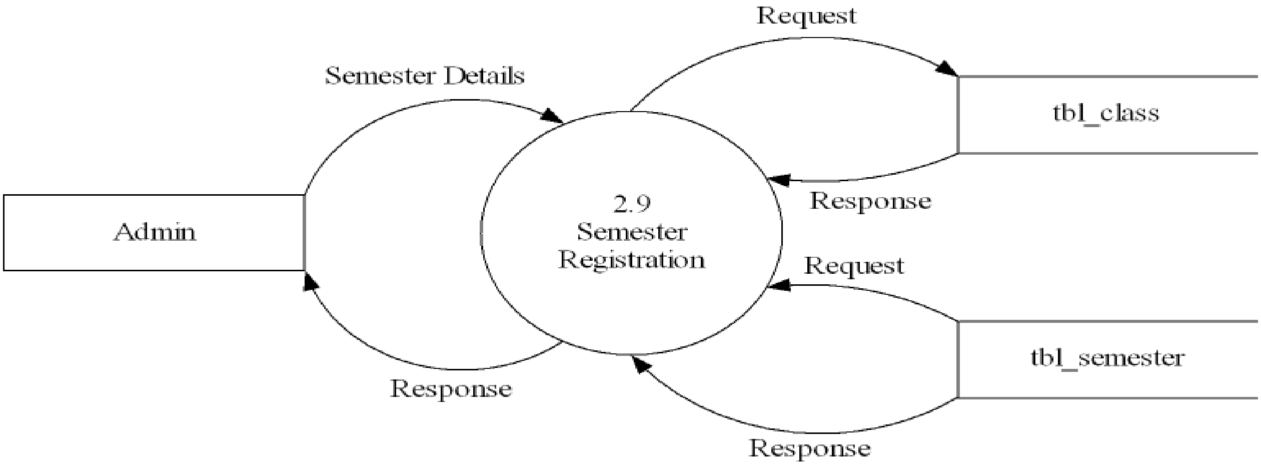


Figure 3.21 Second Level DFD for Semester Registration

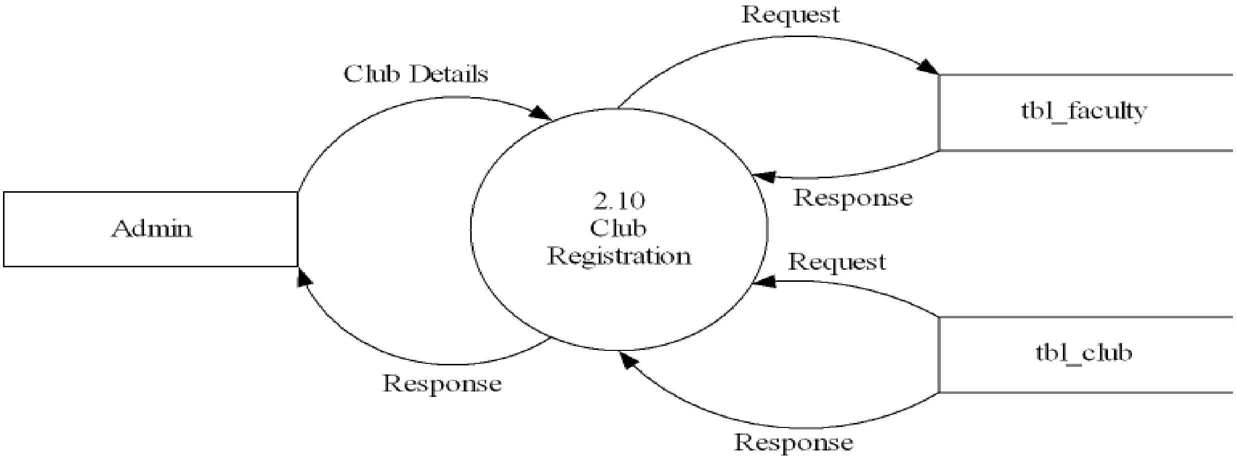


Figure 3.22 Second Level DFD for Club Registration

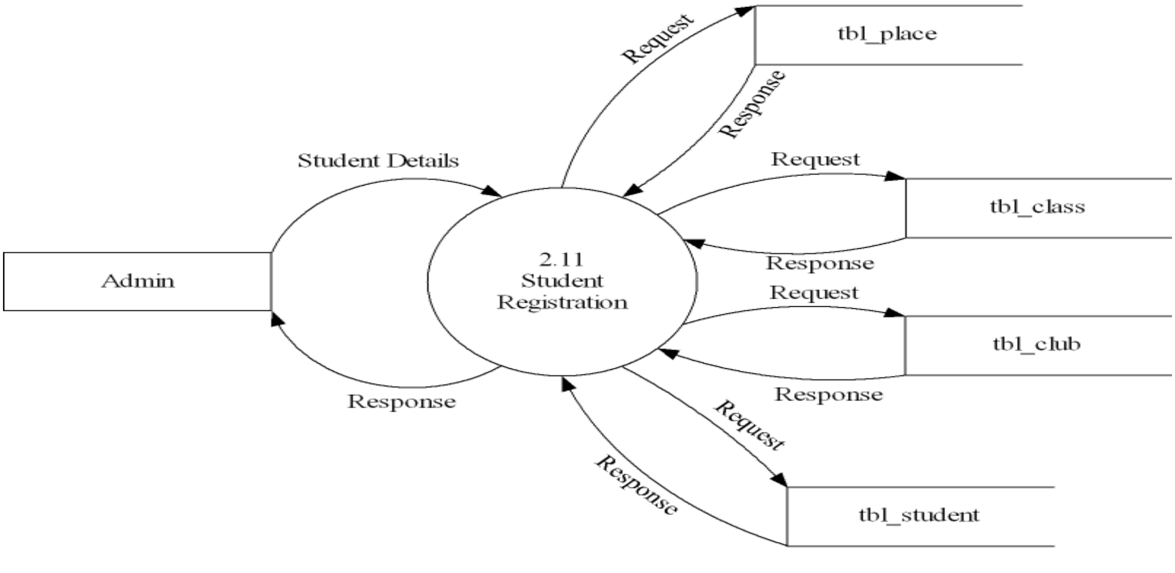


Figure 3.23 Second Level DFD for Student Registration

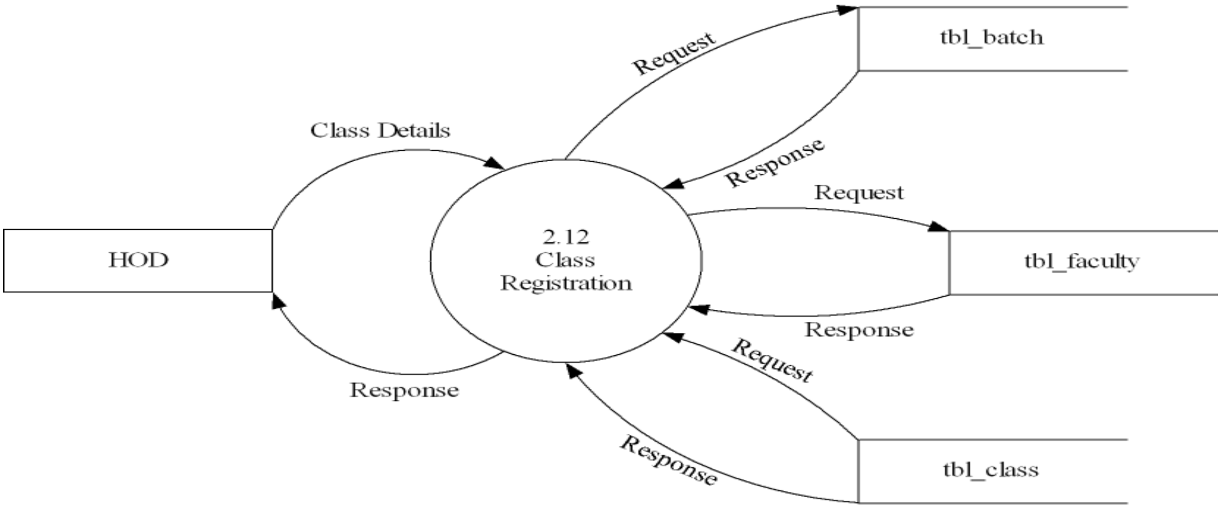


Figure 3.24 Second Level DFD for Class Registration

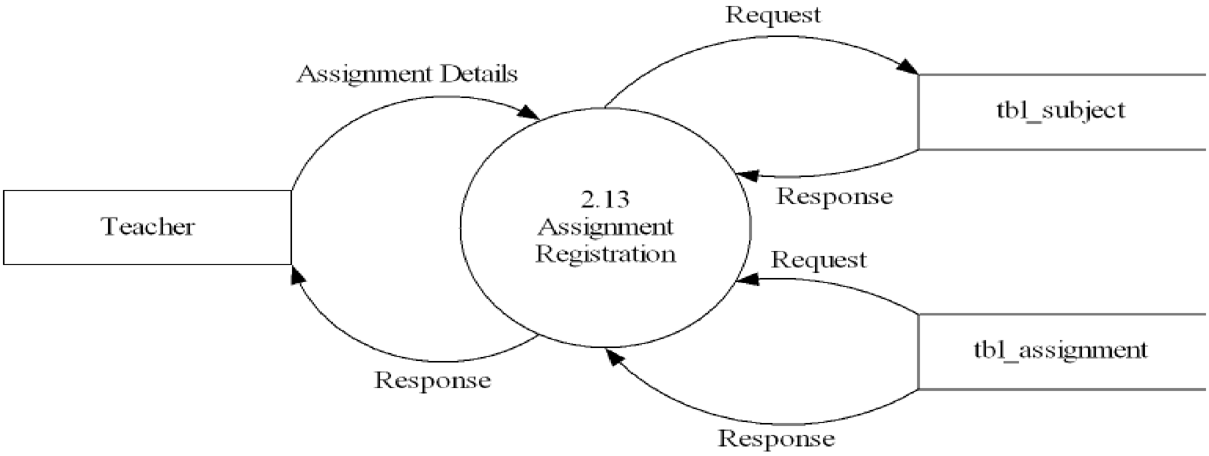


Figure 3.25 Second Level DFD for Assignment Registration

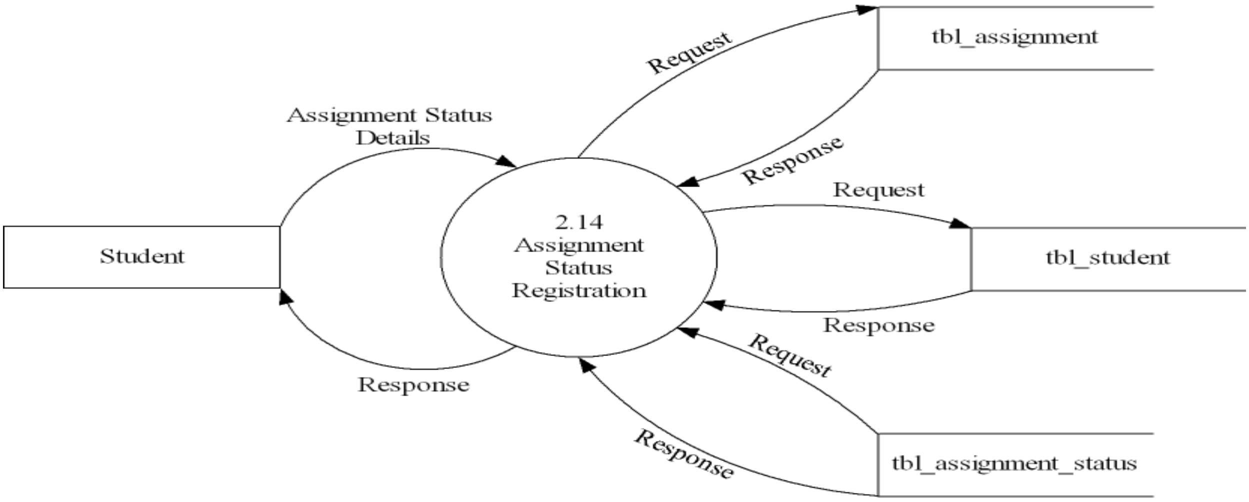


Figure 3.26 Second Level DFD for Class Registration

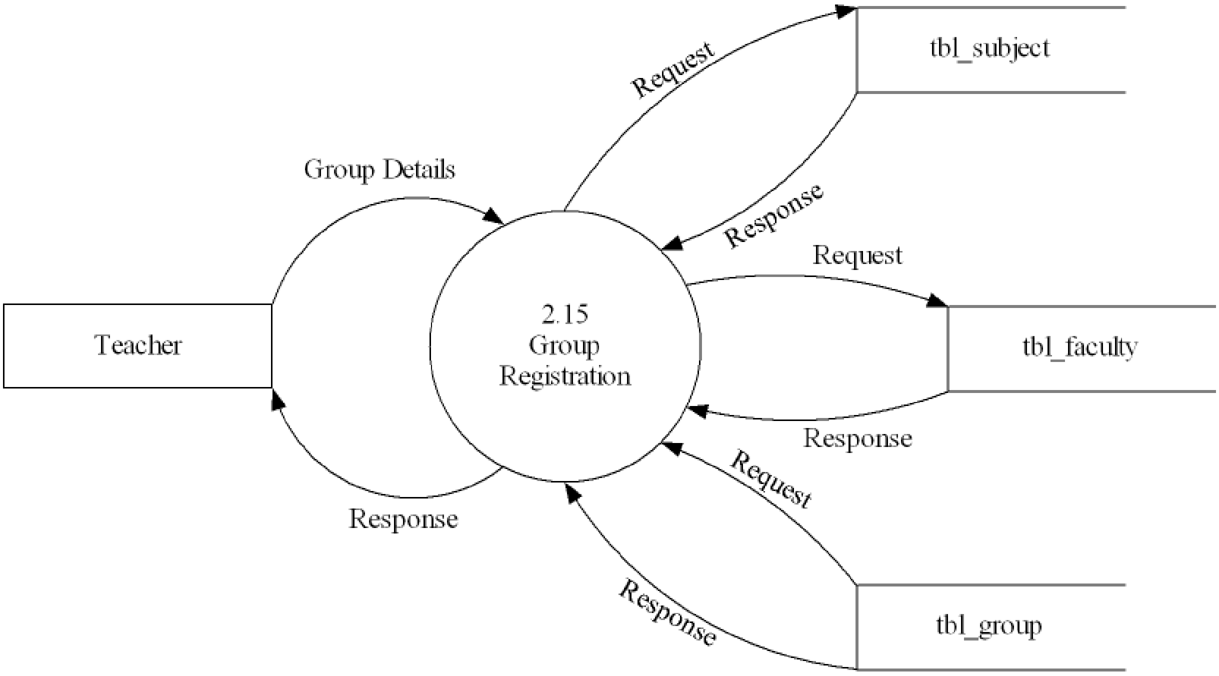


Figure 3.27 Second Level DFD for Group Registration

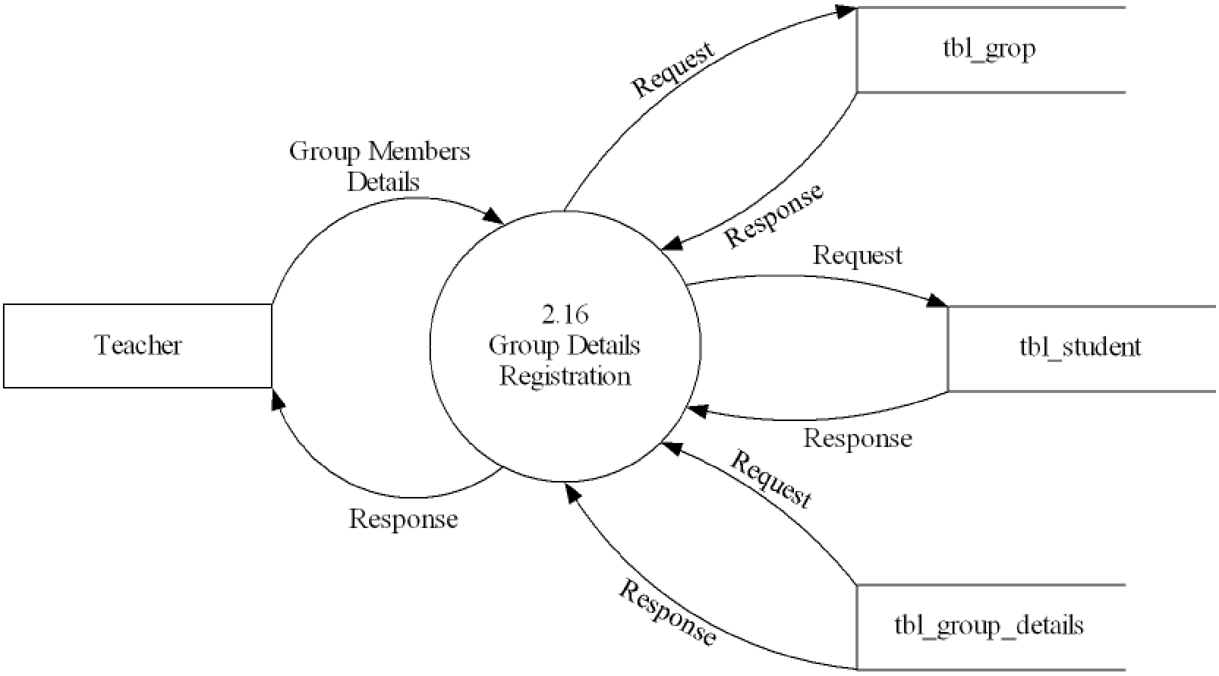


Figure 3.28 Second Level DFD for Group Details Registration

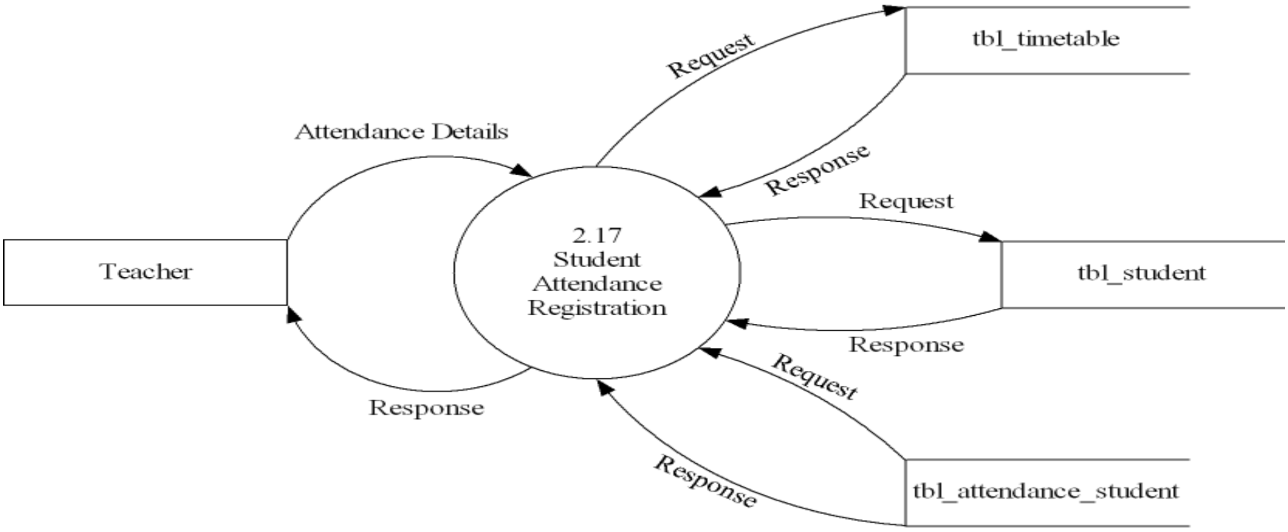


Figure 3.29 Second Level DFD for Student Attendance Registration

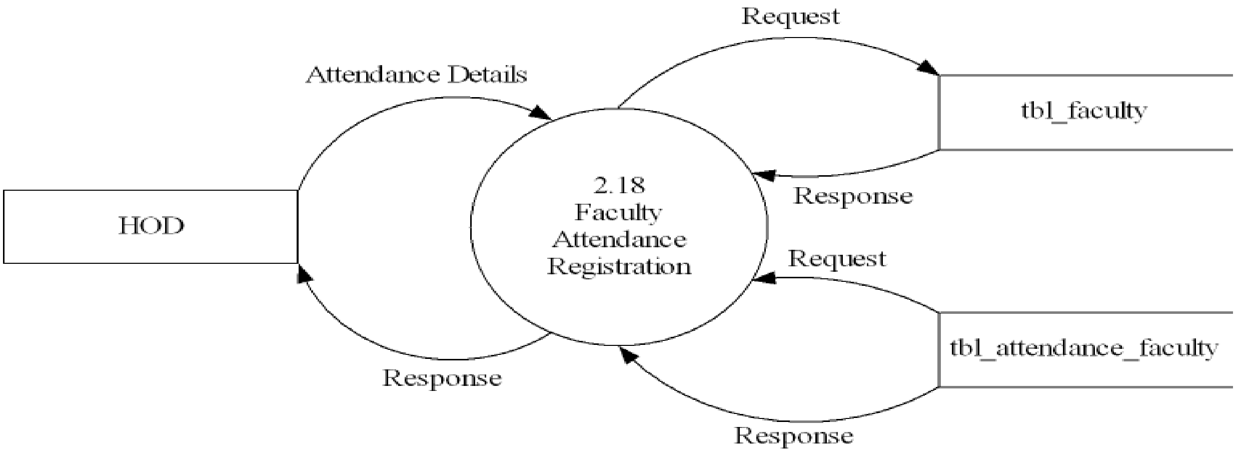


Figure 3.30 Second Level DFD for Faculty Attendance Registration

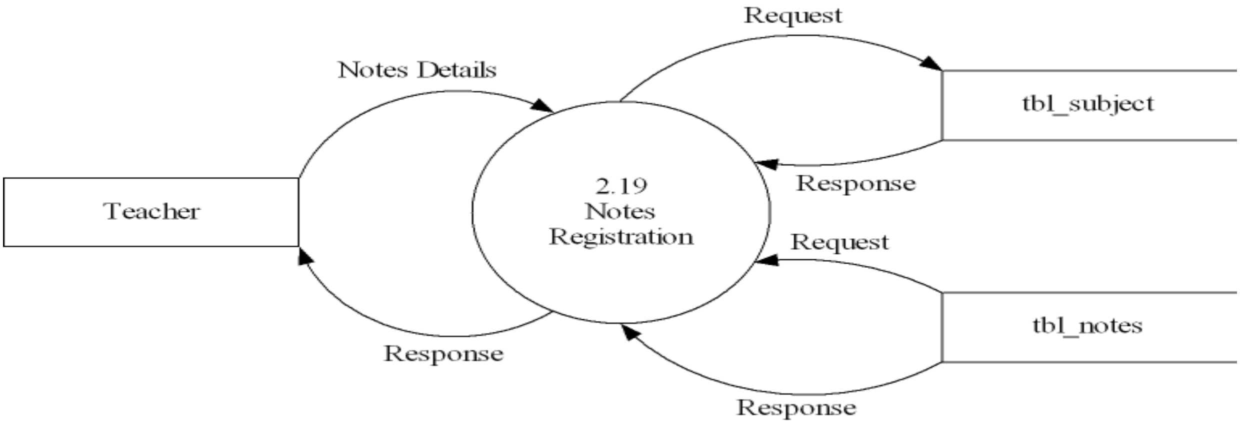


Figure 3.1 Second Level DFD for Notes Registration

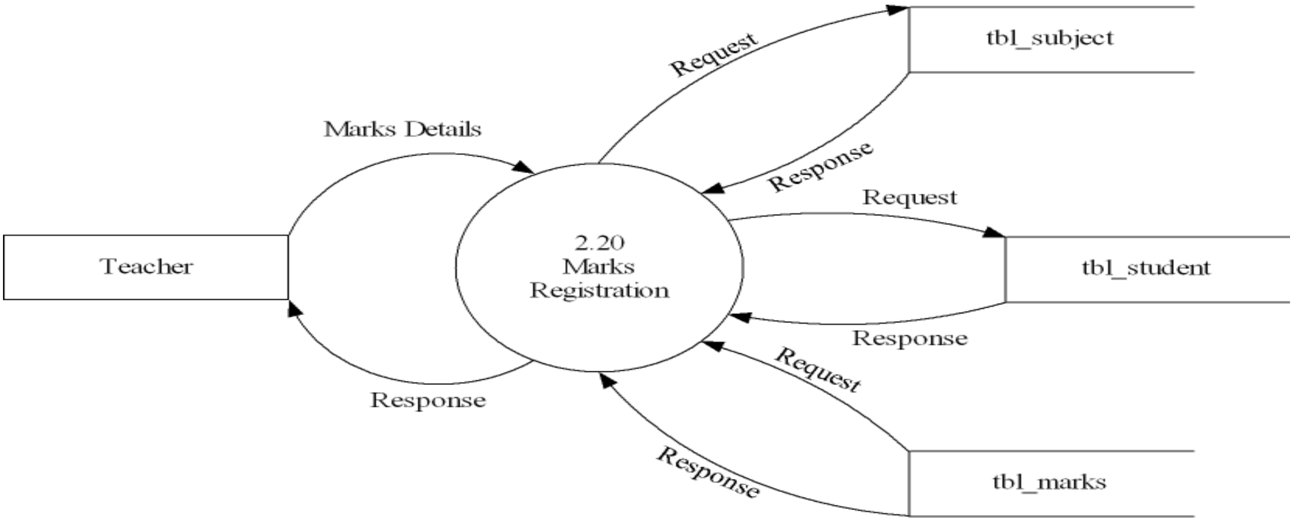


Figure 3.31 Second Level DFD for Marks Registration

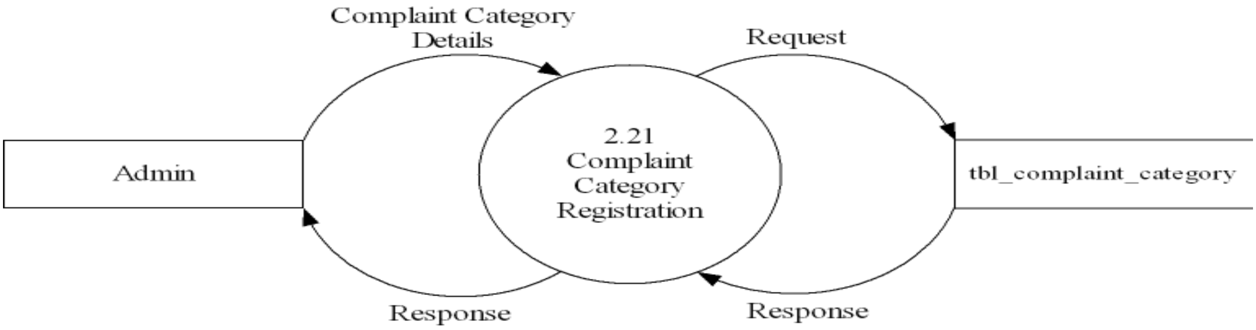


Figure 3.32 Second Level DFD for Complaint Category Registration

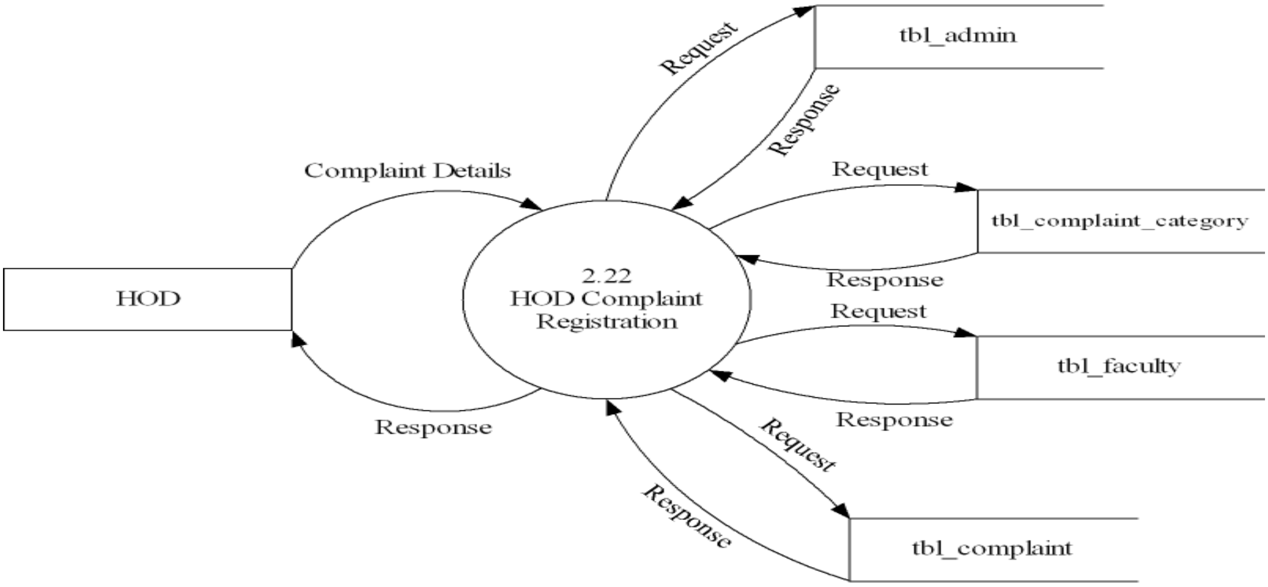


Figure 3.33 Second Level DFD for HOD Complaint Registration

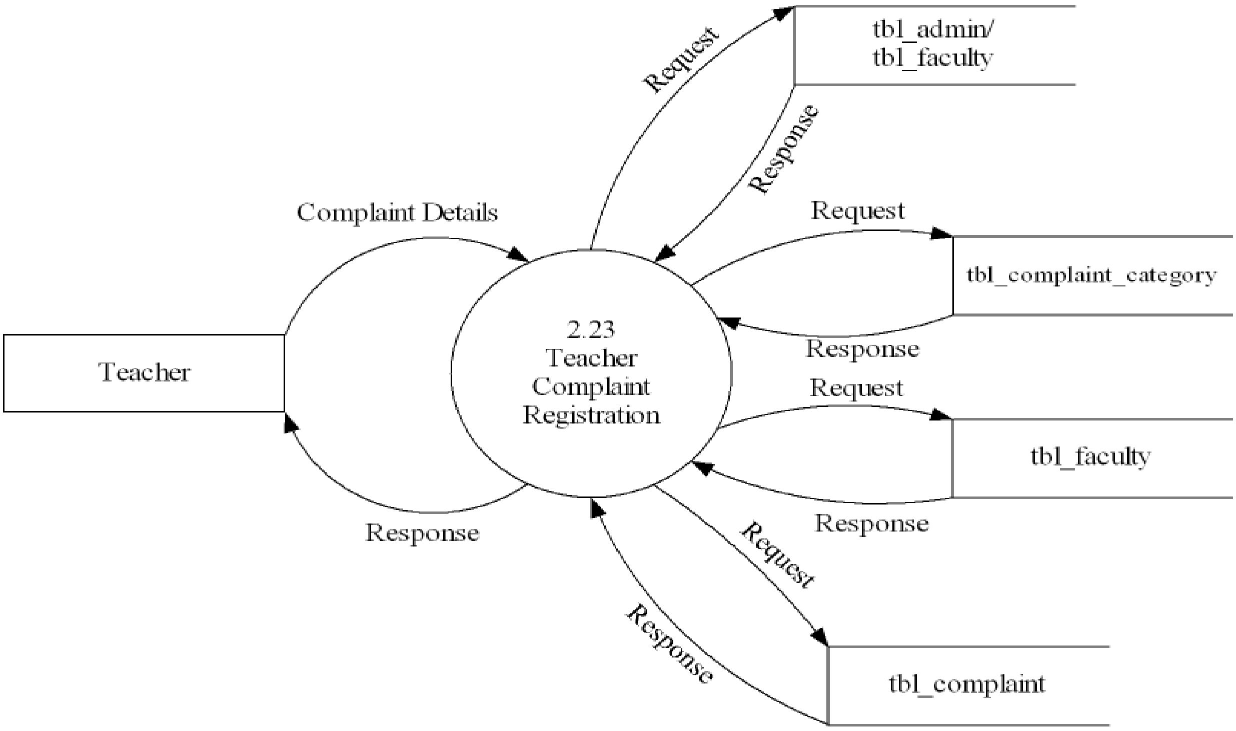


Figure 3.34 Second Level DFD for Teacher Complaint Registration

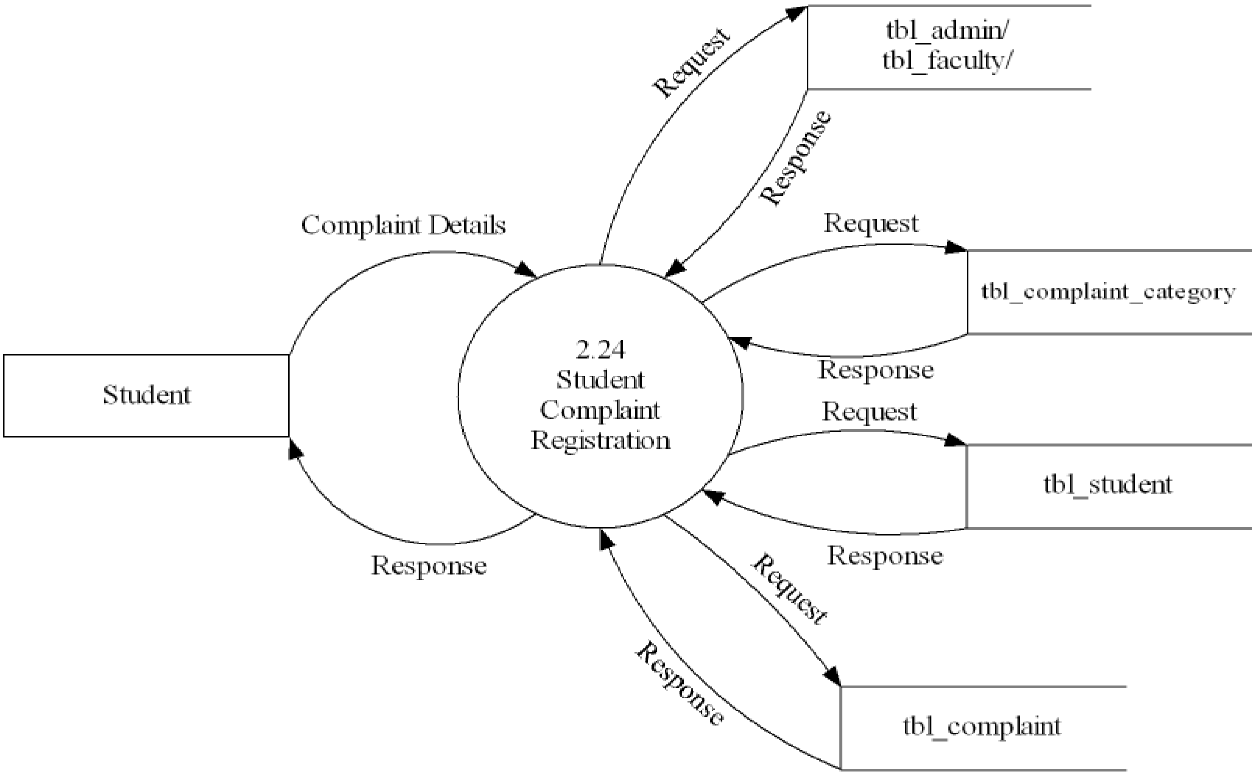


Figure 3.35 Second Level DFD for Student Complaint Registration

Second Level DFD for User Activities

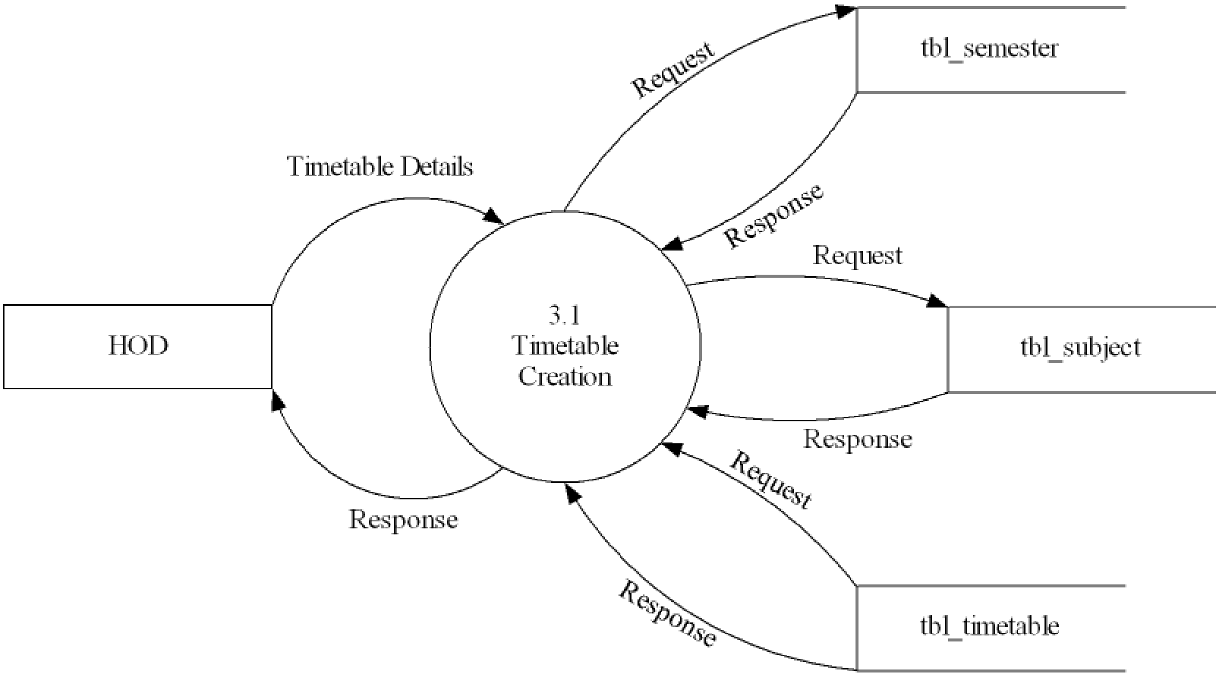


Figure 3.36 Second Level DFD for Timetable Creation

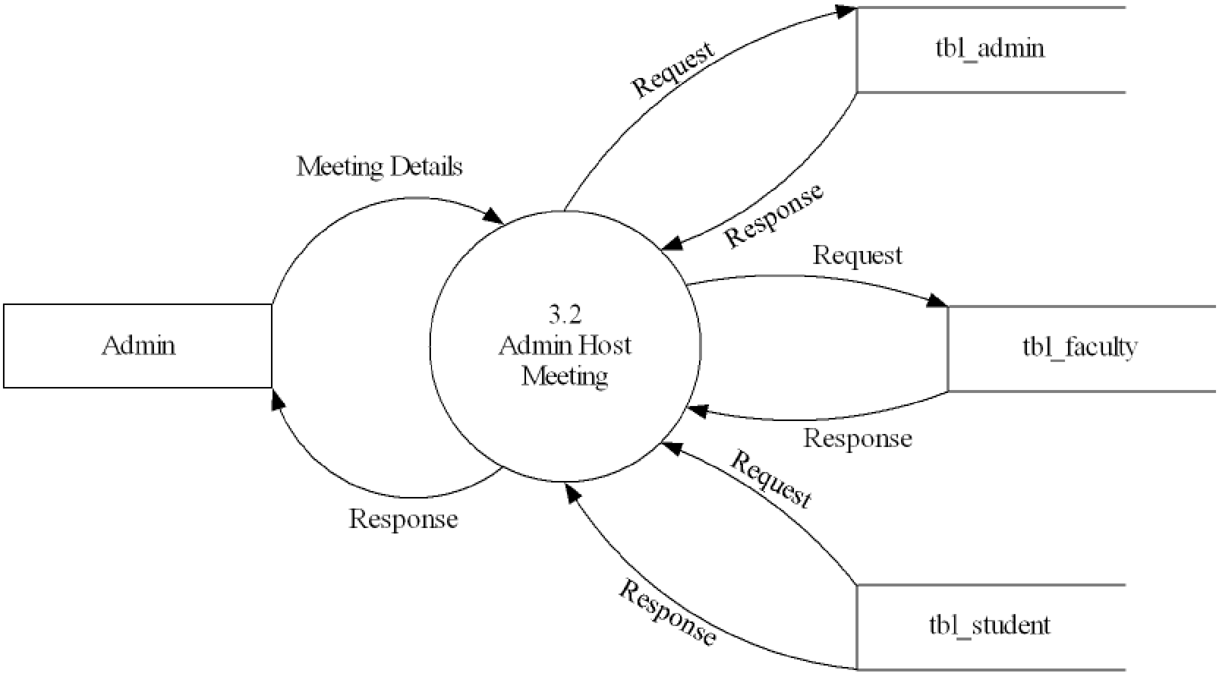


Figure 3.37 Second Level DFD for Admin Host Meeting

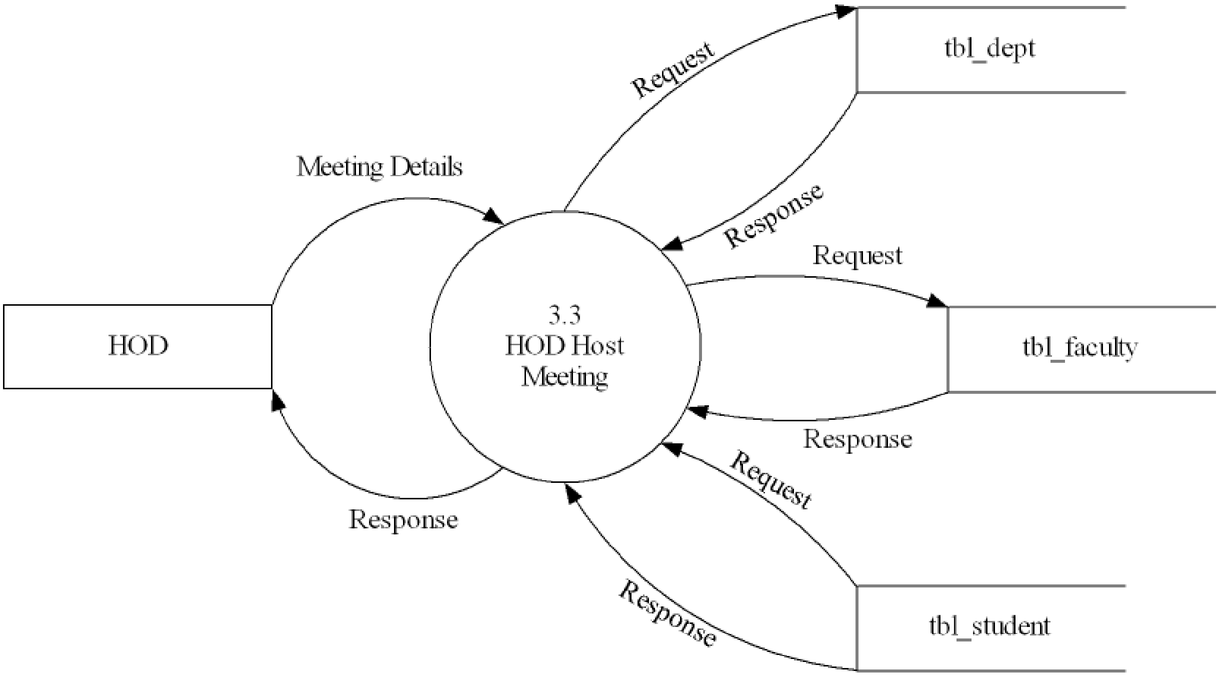


Figure 3.38 Second Level DFD for HOD Host Meeting

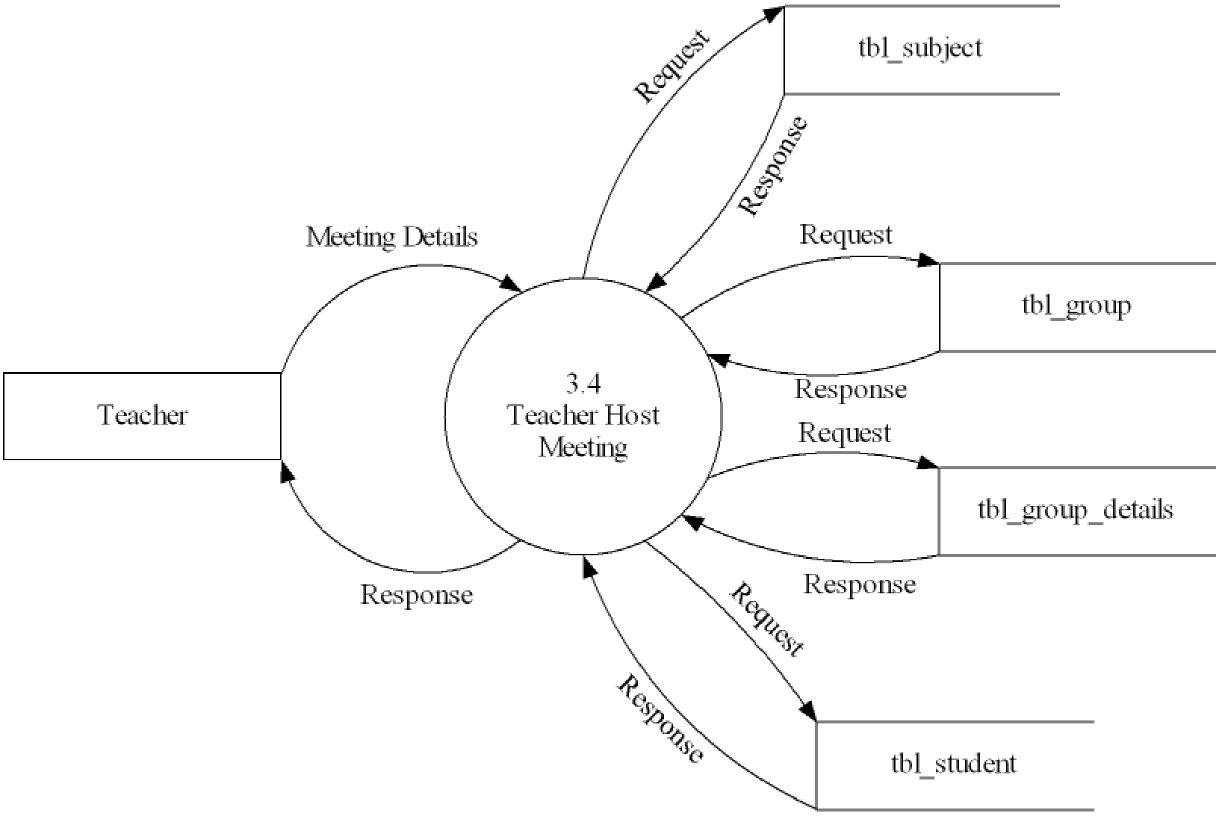


Figure 3.39 Second Level DFD for Teacher Host Meeting

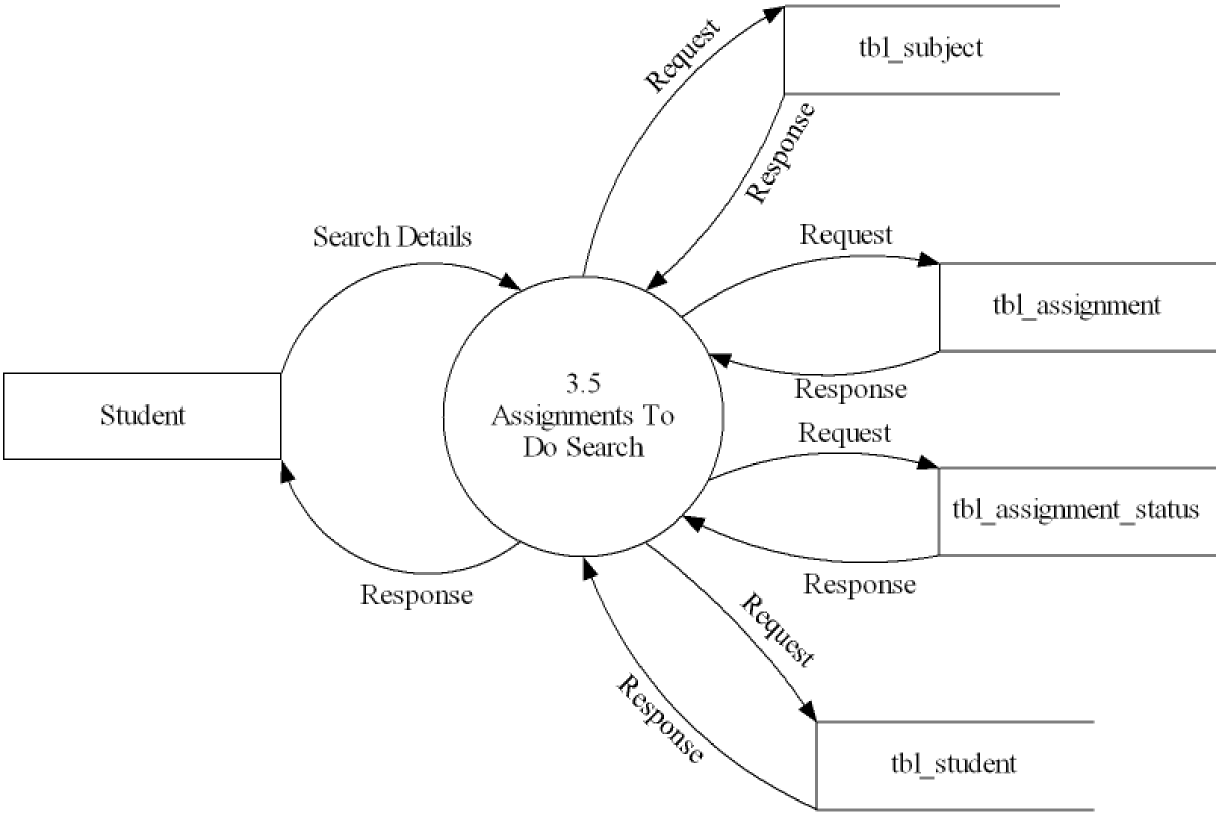


Figure 3.40 Second Level DFD for Assignments To Do Search

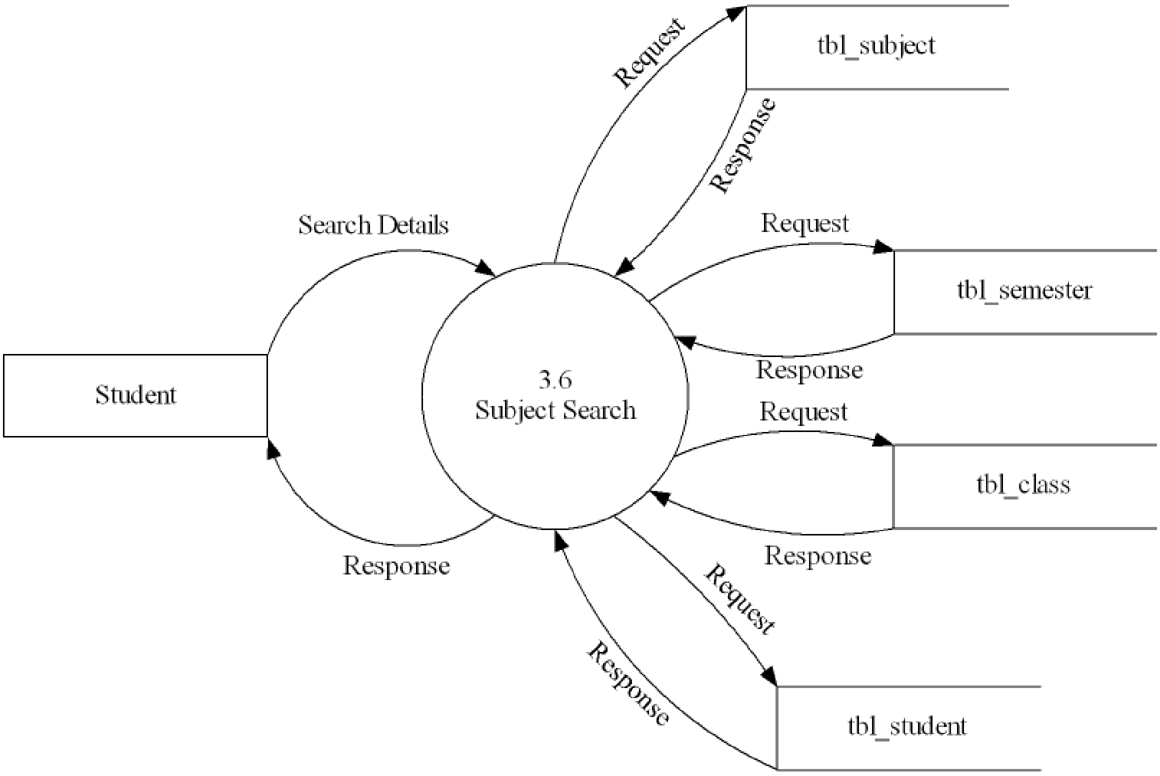


Figure 3.41 Second Level DFD for Subject Search

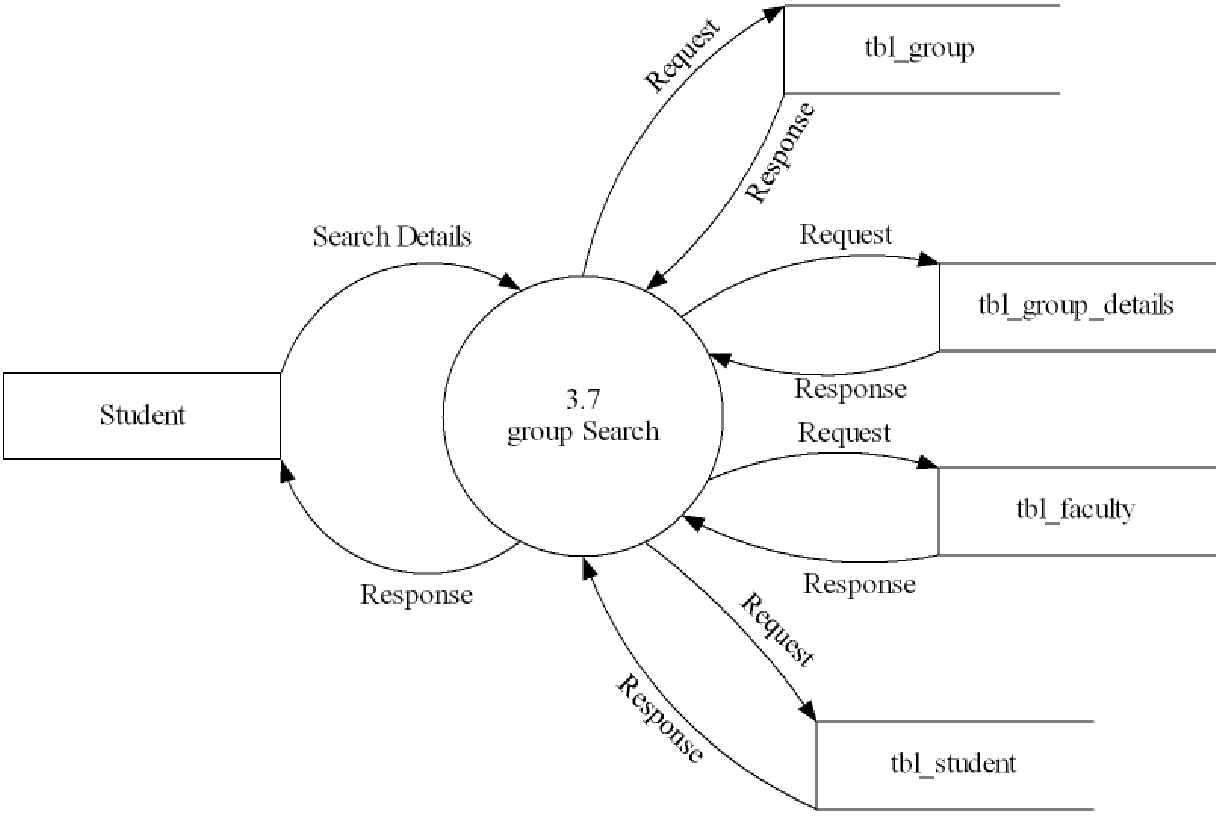


Figure 3.42 Second Level DFD for Group Search

Second Level DFD For Report Generation

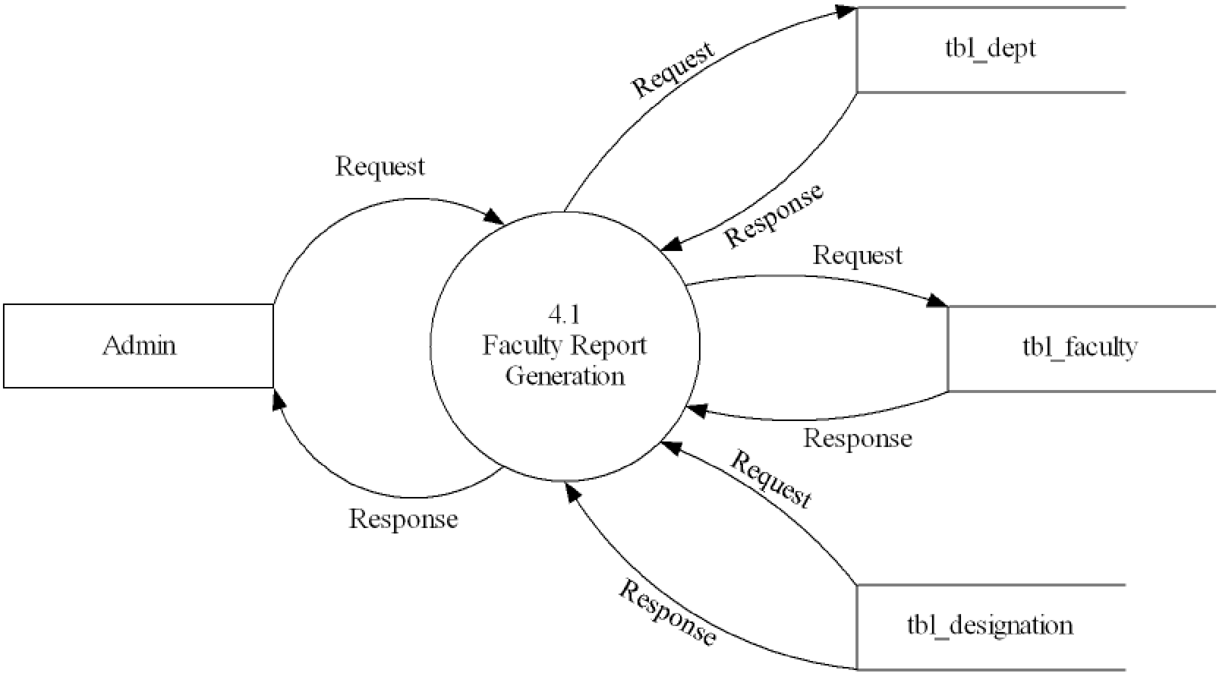


Figure 3.43 Second Level DFD for Faculty Report Generation

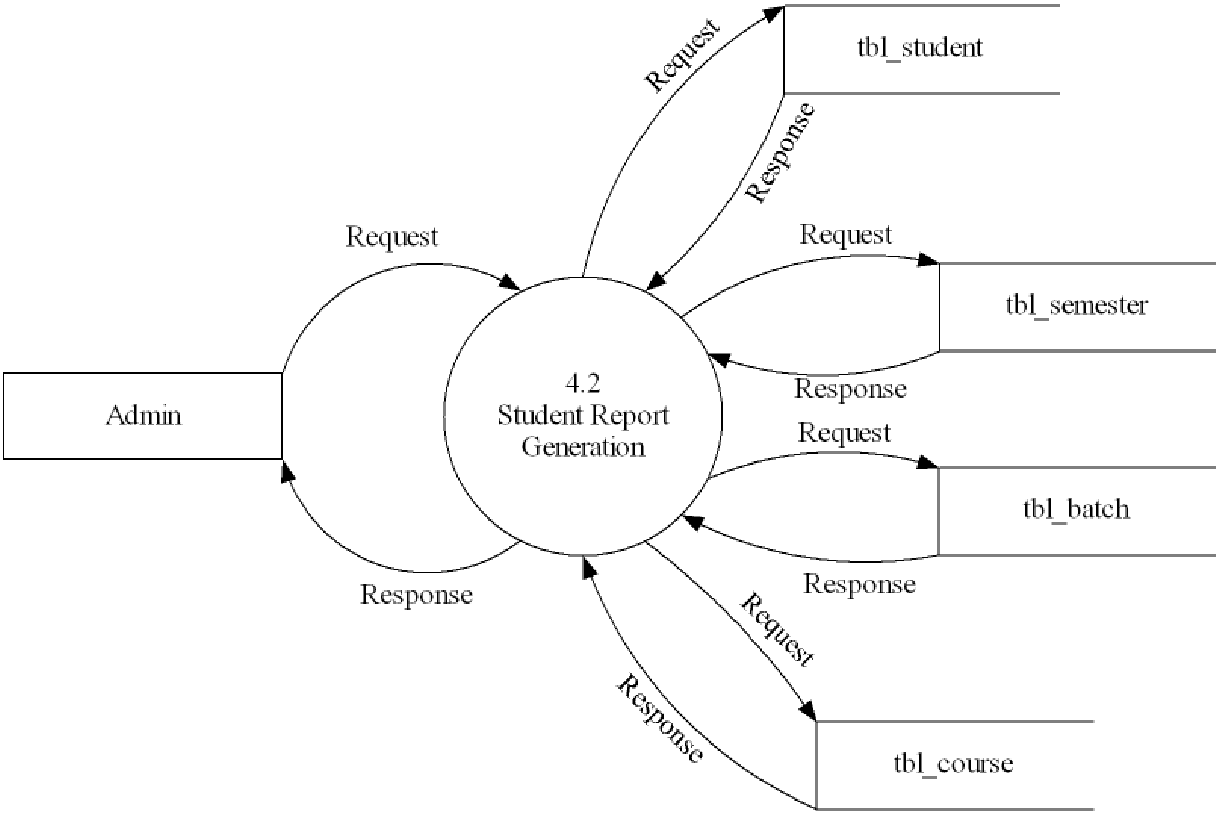


Figure 3.44 Second Level DFD for Student Report Generation

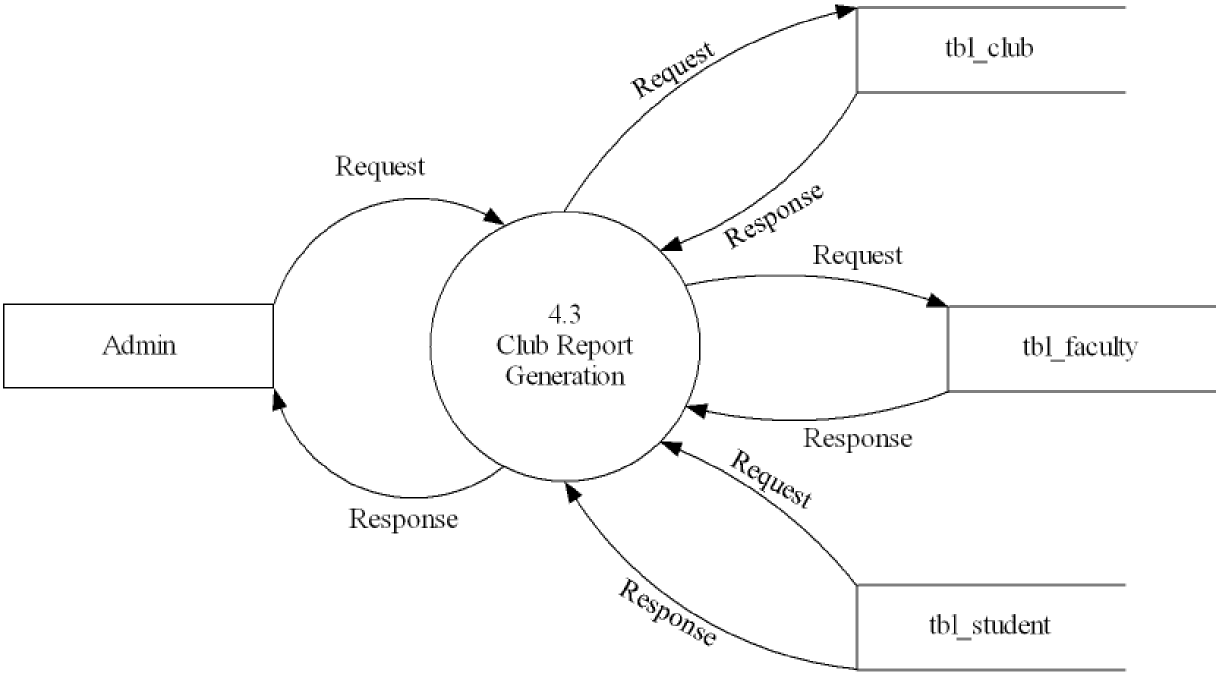


Figure 3.45 Second Level DFD for Club Report Generation

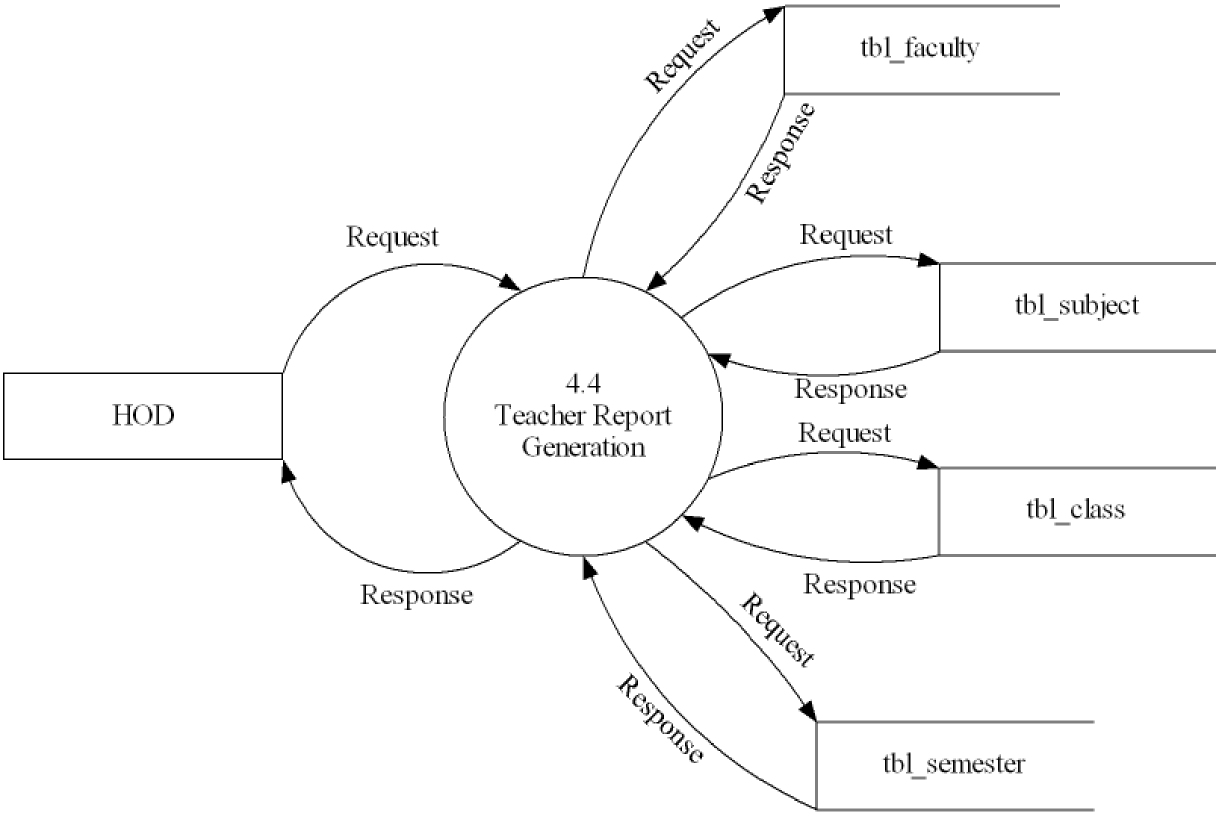


Figure 3.46 Second Level DFD for Teacher Report Generation

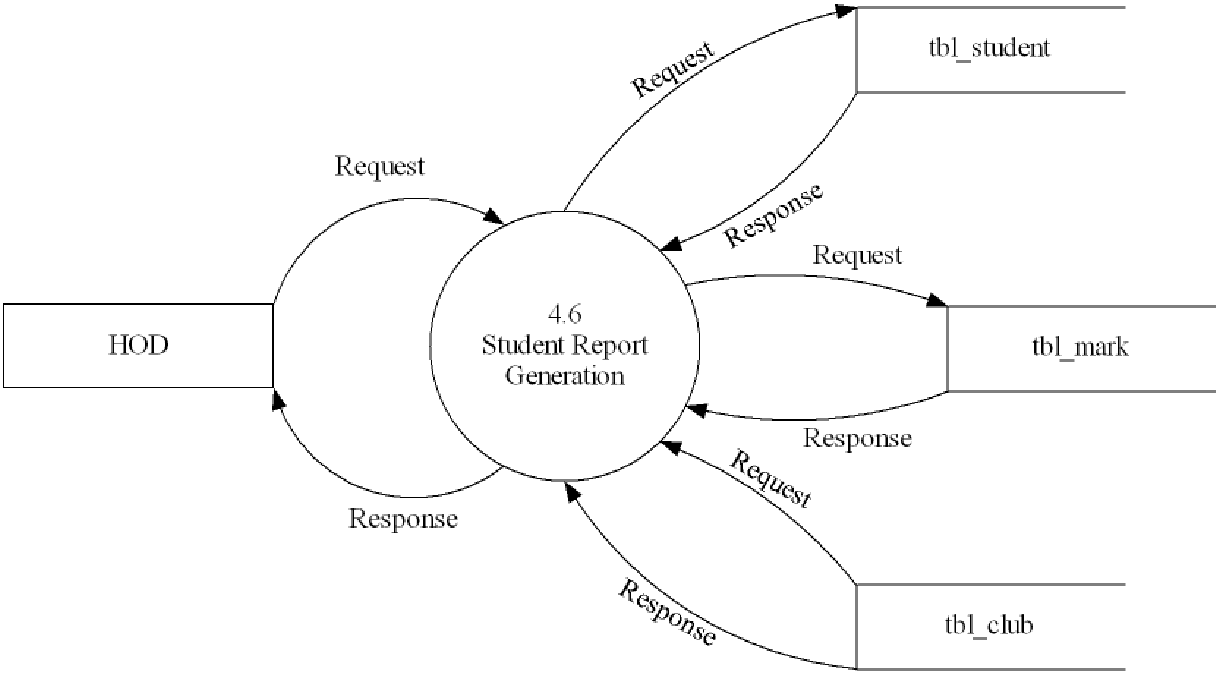


Figure 3.47 Second Level DFD for Student Report Generation

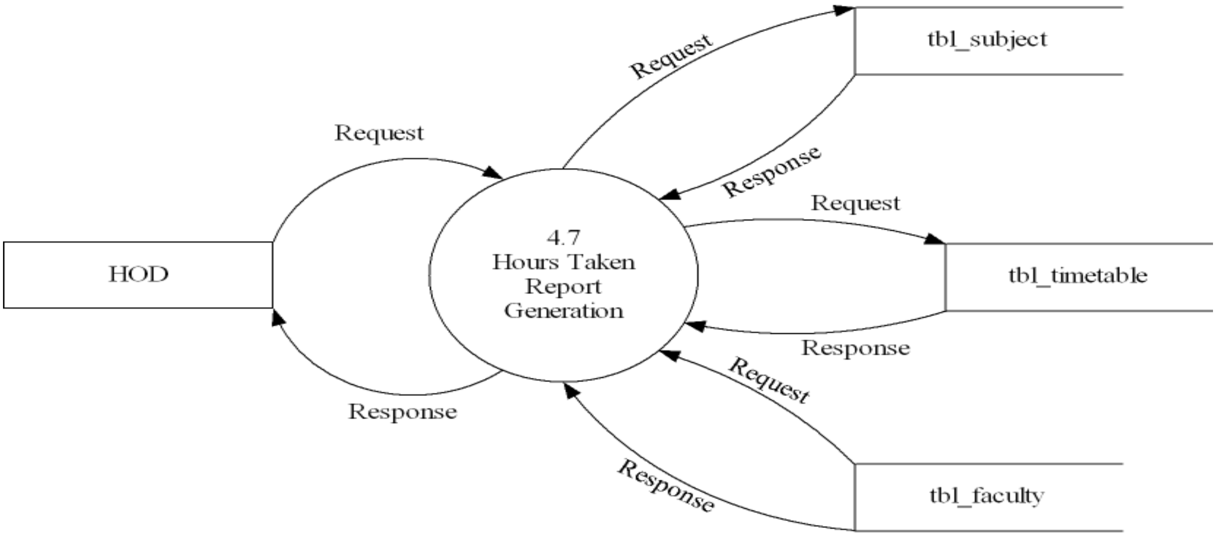


Figure 3.48 Second Level DFD for Hours Taken Report Generation

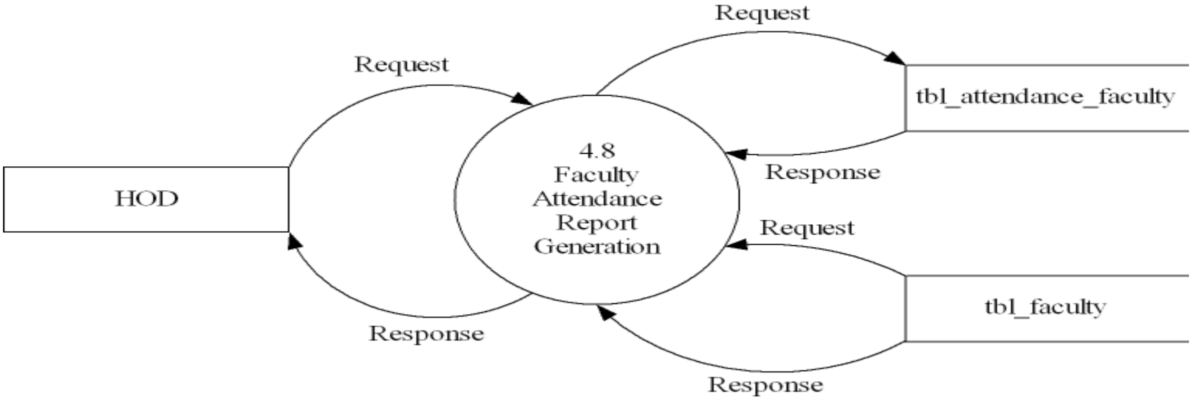


Figure 3.49 Second Level DFD for Faculty Attendance Report Generation

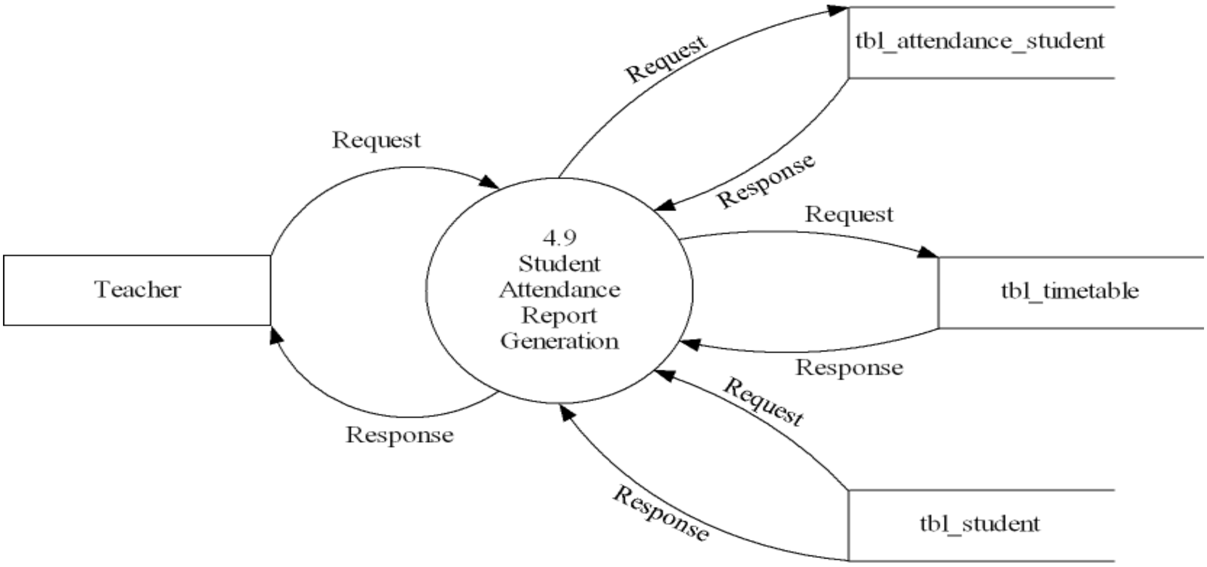


Figure 3.50 Second Level DFD for Student Attendance Report Generation

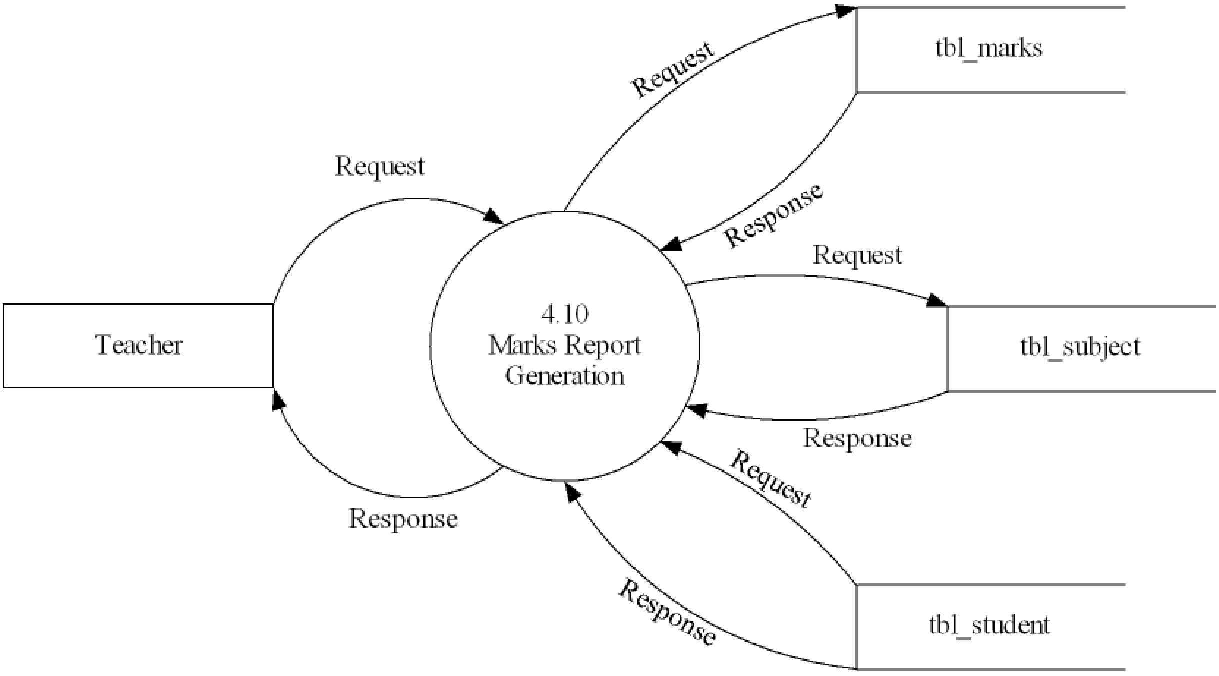


Figure 3.51 Second Level DFD for Marks Report Report Generation

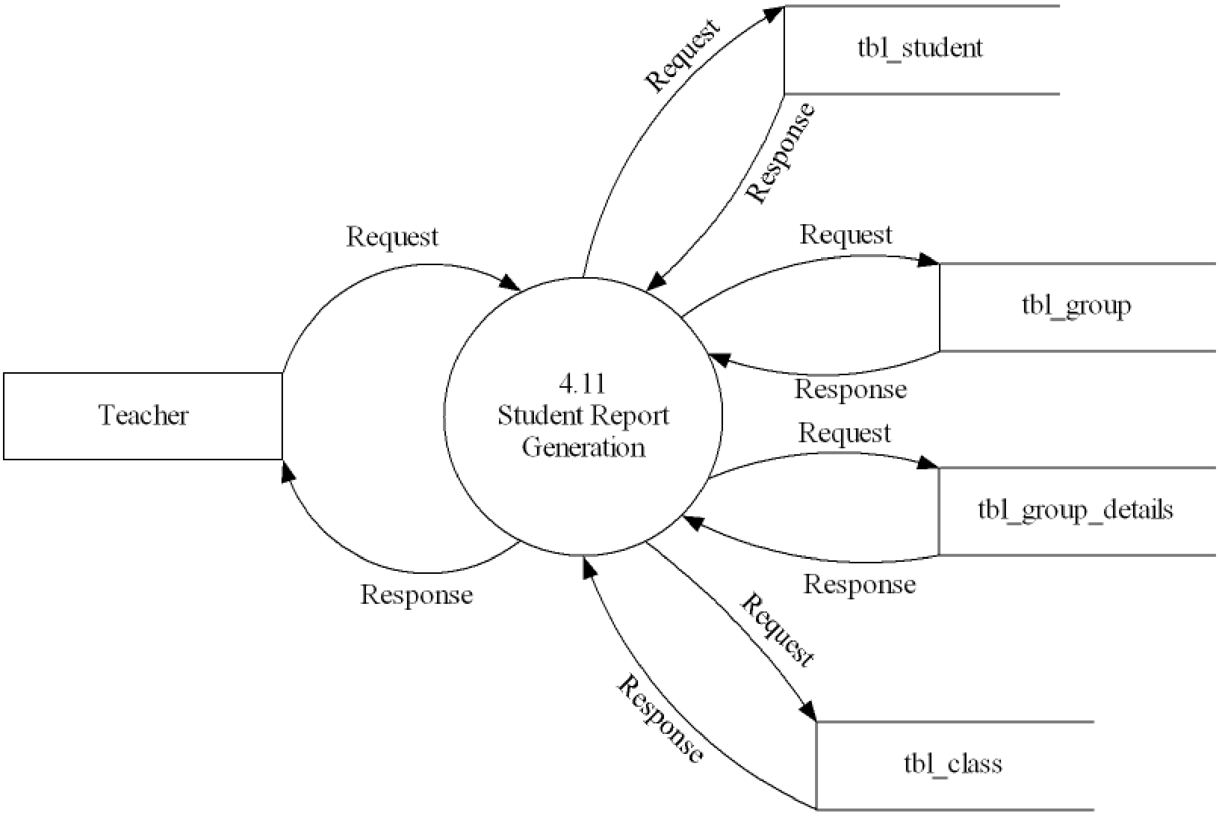


Figure 3.52 Second Level DFD for Student Report Generation

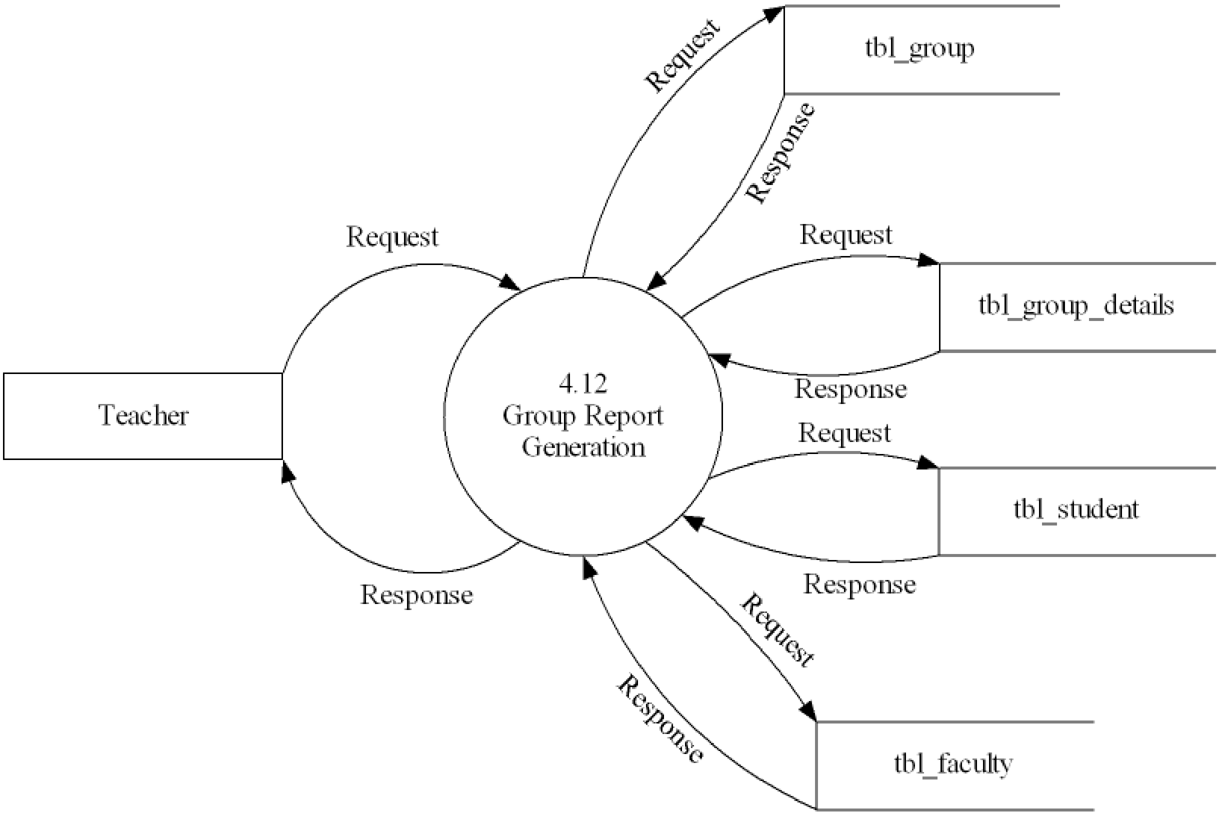


Figure 3.53 Second Level DFD for Group Report Generation