Department of Computer Science & Engineering
The University of Texas at Arlington

Architectural Design Specification
CSE 4316: Senior Design I
Summer 2018

Acorn Interactive Systems
Interactive Degree Planner

Ben Pham
Jacovo Fernandez
Julian Cisneros
Natnael Kebede
Octavio Garcia

Revision History

Revision	Date	Author(s)	Description
0.1	08.09.2018	BP,JF,JC,NK,OG	document creation

CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

The degree planner will allow students to visualize class schedules for their upcoming semesters. The concept is a degree planner that students can use to facilitate the task of determining how their semester should be structured in regard to the following.

1. Courses needed to meet degree requirements.
2. Number of classes that will be taken.
3. Difficulty of these classes.

After evaluating these three variables the degree planner will lay out possible schedules within the scope of the Computer Science major. The students will then have the ability to change the classes of the proposed schedule and the degree planner will verify these changes against class requirements to either allow or deny the new schedule.

2 System Overview

The system will be comprised of three layers. These layers will handle interactions with the users, take in data required to generate the schedules and use logic to verify course requirements against university standards. The communication will be bidirectional within these three layers, allowing each layer to send and receive data from other layers.

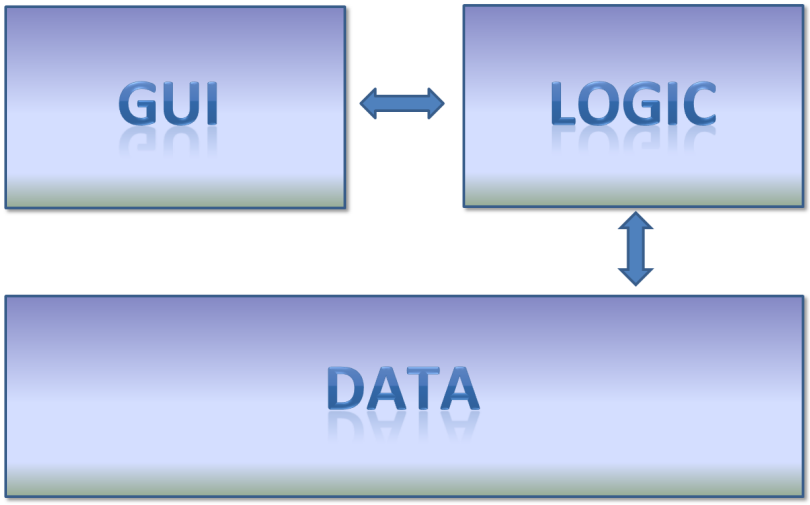

Figure 1: Architectural Layers diagram

2.1 GUI Layer Description

This layer is responsible for all interaction with the users of the system. The GUI layer will contain the main method where program execution will start. At the start of the program this layer will be responsible for showing the user a welcome message along with a description of what the program could do, along with helpful tips. This layer will also accept input from the user and send this data to the logic unit for processing, then once this data has been processed the GUI layer will show the user a new screen based on their responses to the previous questions. The subsystems included are input/output, screen components, menu and screen layers.

2.2 Data Layer Description

The data layer will have the following three subsystems: database, save, and load. The three major functions of this layer will be to communicate with the system database, save current work and load any previous work if the user wishes to continue from a saved session. At the beginning of the execution, the data layer will upload all course information to a vector. This vector will be housed in the logic unit, where it will be manipulated according to the users response to the system inquiry.

2.3 Logic Layer Description

All the system logic will be housed in this layer. The logic layer will be responsible for verifying individual course requirements as well as course difficulty distribution. This layer will communicate with both the GUI and Data layers to guide the user through the use of the system. The user answers will be computed in this layer with the results of these computations being the output viewed by the user.

3 SUBSYSTEM DEFINITIONS & DATA FLOW

The application software system is made up of many subsystems under each top layer as described in previous section. They are the subsystems with specific capability that function independently and work in harmony to support the layer in performing the designed role.

Starting with the data layer, course information is first retrieved from the database and flows through data controller. Upon Users request, data can be transferred to saving subsystem for data retention on a removable media. On other hand, data could also be retrieved from a removable media through the loading subsystem. The layer is connected with the logic layer by way of a bidirectional flow of data and control path.

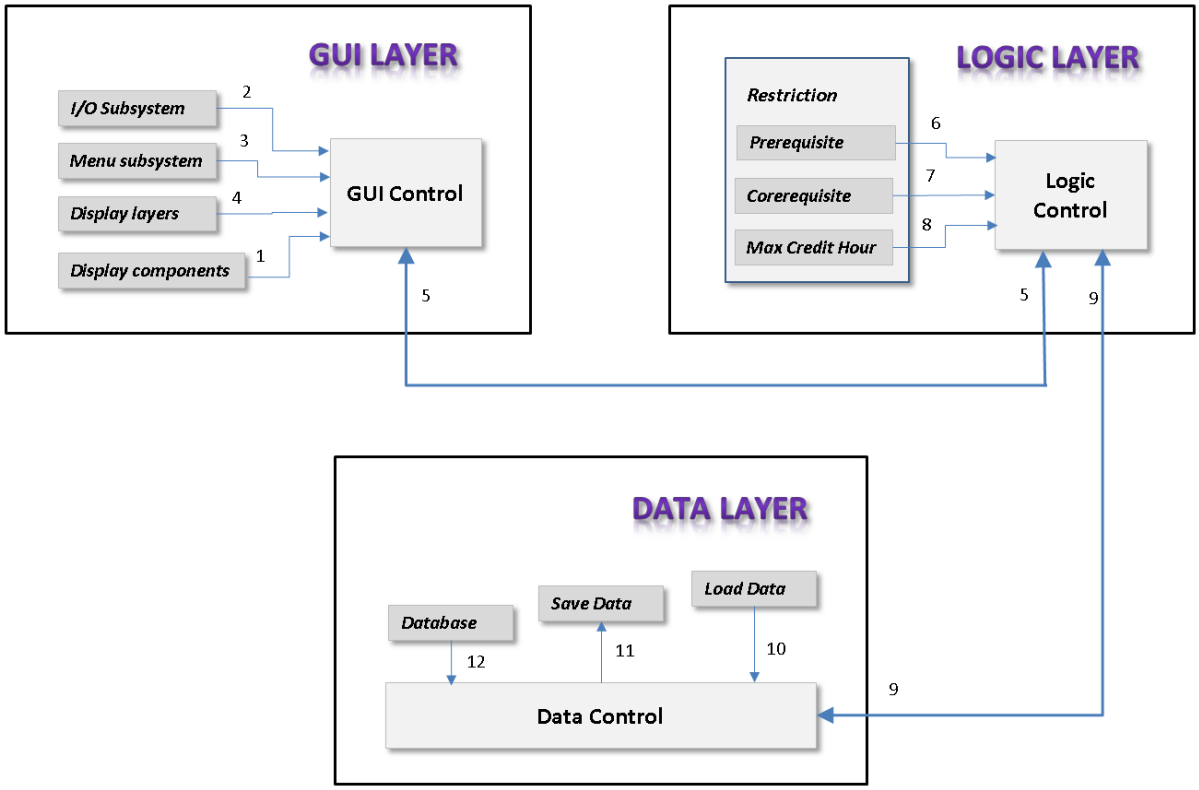

Figure 2: Data flow diagram

In the logic layer, the logic controller is the main processing place where all data and control information are traveling through. It processes data received from data layer and passes over the information to the GUI layer for displaying. The data can travel back in the opposite direction when user input data is received. Likewise, data fed to the GUI controller is also from restriction subsystem where constrains information is provided to assist data processing.

Within the GUI layer, all data flows from the logic layer are presented on screen using display components, menu subsystem and display layer control subsystem. These elements are capable of receiving user input information such as a mouse click event on checkbox control that indicates user selection on an item , this data can be sent back to the GUI controller for further processing.

4 GUI Layer Subsystems

The Graphical User Interface layer will contain all the sections that primarily focus on displaying windows and menus to all users. The GUI subsystems allow for a professional environment for users to interactively build and create their degree plan for their majors. Each GUI window will be responsible for different tasks and they will each have their own specific display components. Input and output are handled on the GUI components as well as on menus. All sections of the GUI layer will be handled primarily on the GUI control subsystem.

4.1 Display Components

Each Display will have specific components provided by the GUI control which involves the use of buttons, check boxes, menus and other various types of methods for user interfaces.

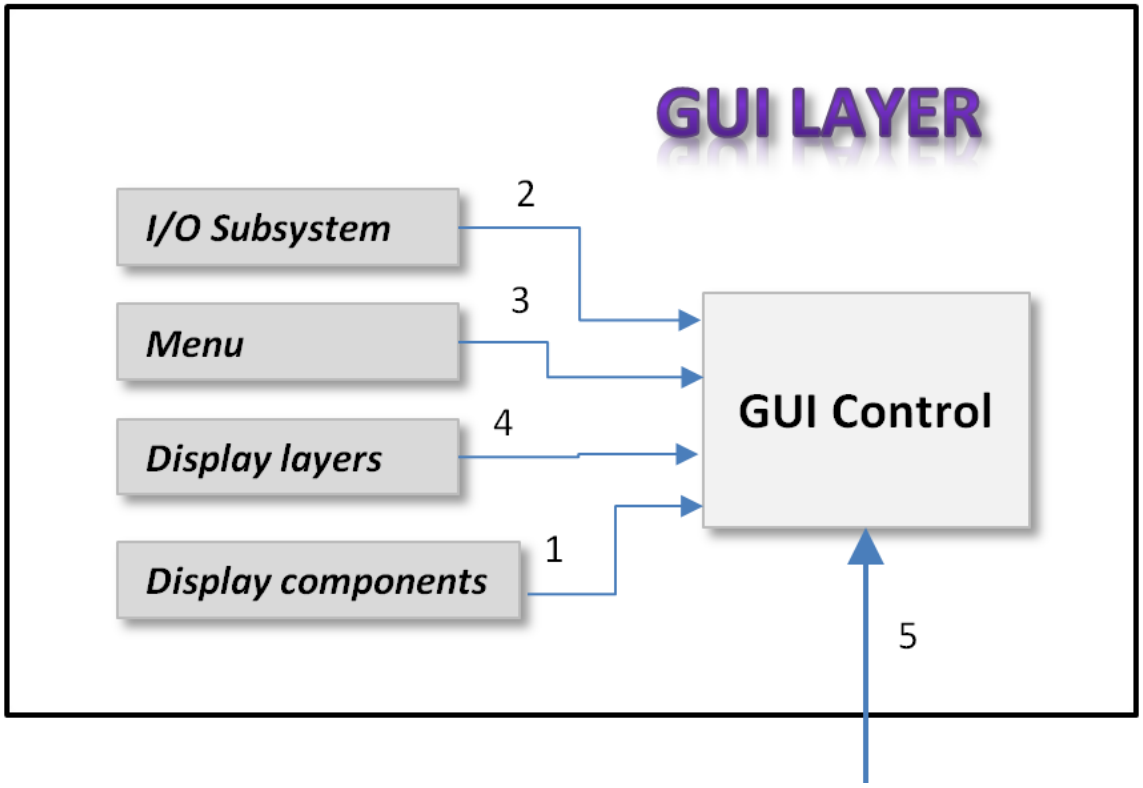

Figure 3: GUI layer and subsystems diagram

4.1.1 Assumptions

Java Swing allows for a variety of screen components such as buttons and check boxes. The display components will be used to make the system more professional and allow the user for interactive use.

4.1.2 Responsibilities

The responsibilities of the display components will be to provide the user with an interactive environment where a user can use provided buttons, check boxes, and other menu options to build their degree plan. Display components will work together with multiple display layers while taking in various forms of input and providing users with expected outputs.

4.1.3 Subsystem Interfaces

Screen Components are connected to the GUI control where most of the implementations for display functions will be passed.

Table 2: Display Components Subsystem interfaces

ID	Description	Inputs	Outputs
#1	Screen components are tailored and controlled within the GUI control subsystem.	Provided components will be buttons, check boxes and various menus that need to be clicked.	Clicking each component registers the action within the program and allows the user to progress on building their degree plan.

4.2 Display Layers

The Display Layers will be consist of separate windows interfaces that handle different tasks. The GUI Sections will be implemented using different layers so that each user will have the ability to handle different tasks within the product.

4.2.1 Assumptions

Each interface will have different properties and components when creating a degree plan. Only one display window will be open at any given time allowing for the user to focus simply on the window that is currently being displayed.

4.2.2 Responsibilities

The responsibilities of the display layers are to provide the user a window to perform operations and interactive create a degree plan. Additionally, display layer will have different windows for different operations, whether it is selecting classes or manipulating an already built schedule.

4.2.3 Subsystem Interfaces

Display layers are connected to GUI control so that each display that is shown, is strictly managed within the GUI control subsystem.

Table 3: Display layers Subsystem interfaces

ID	Description	Inputs	Outputs
#4	All display layers are managed within the GUI control subsystem. This will restrict the displays to only have one window open at any given time while properly managing all types of layers that are to be displayed throughout the program.	Display windows will only have inputs that allow for exiting, minimizing, or full screen options. These are meant to be a window that will hold all the other sections of the GUI subsystems.	The display window will simply minimize, exit, or full screen when the specific button is pressed on the display window itself.

4.3 Input - Output Subsystem

Graphical user interface sections will have their own inputs and outputs for corresponding tasks. These tasks are handled by various forms of system callbacks that are used to aid users so that they may create their personalized degree plans.

4.3.1 Assumptions

The assumptions based on input and output would correspond to how each display component reacts when there is a specific form of user interaction. Input and output on each GUI section will all be assumed to have a specific task that is provided to aid the user while they interactively build their schedule.

4.3.2 Responsibilities

The responsibilities of input and output involve registering multiple forms of callbacks. When a button is pressed, the user would assume for an action to occur. This is handled by the input and output callback system. All GUI displays will have specific inputs and outputs for user interactions and they will be tailored to do simply what a user would expect to occur.

4.3.3 Subsystem Interfaces

Input and output are connected and handled within the GUI control subsystem so that further implementation of the callback system may be pursued within the program itself.

Table 4: Input-Output Subsystem interfaces

ID	Description	Inputs	Outputs
#2	Inputs and outputs are callbacks that are handled within the GUI control subsystem, hence the connection.	The inputs will vary depending on the type of component. Whether it is pressing a button, check box, menu buttons, each component will have a callback when clicking.	If a button is clicked, then an action will occur. Check boxes will register input to be used for later, help menus will display instructions, save will register all the data to a file to be used later, load will upload a saved file to continue building a degree plan and exit will simply exit the program.

4.4 GUI Control Subsystem

The GUI Control will have the purpose of combining all the Java swing components for display purposes as well as sending data requests for usability or approval.

4.4.1 Assumptions

The GUI Control will not take part in any data manipulation, but instead simply process requests and data to the appropriate subsections. Java swing will be used to control all the components required to display it to the user.

4.4.2 Responsibilities

The main responsibility will be controlling and managing interaction screens which consist of windows layers and Java swing components. The second purpose is to act as a mediator between the subsections within its own layer and the Logic Control subsection.

4.4.3 Subsystem Interfaces

The input for this subsystem is data comes from all of the other subsystems within the GUI layer in the form of actions and events. The output is data made toward the Logic Control which come in the form of data requests or guideline consistency checks.

Table 5: GUI Control Subsystem interfaces

ID	Description	Inputs	Outputs
#5	The GUI Control subsystem will receive the confirmation status on the requested action/event from the GUI layer. The input can also take form as raw data passed from the Data layer	The input will take a form as either a confirmation if there was a request, or data	The output will take form as an action/event request from the other GUI subsections

4.5 Menu Subsystem

This subsection will contain program options which allow the user to save/load their progress or seek more helpful information on how to use the program.

4.5.1 Assumptions

The selectable options will use components gathered from the Java swing library.

4.5.2 Responsibilities

The responsibility of the menu system is to provide access to application-specific features at any time while the application is running. Primarily, the most relevant features will be loading/saving progress and obtaining help information.

4.5.3 Subsystem Interfaces

There will be a one-way interface to the GUI Control which allows the selected menu item triggering event to be routed to its appropriate functions within this subsystem, which handle and react with a proper response.

Table 6: Menu Subsystem interfaces

ID	Description	Inputs	Outputs
#3	The Menu subsystem will be the bar that sits on the top of the screen which will allow the user to load and save progress points, or request a help screen	The input will be the request for an event or action which is triggered by the user selecting an option.	The output will depend on the option that the user selects. It can either take the form as a new window where a file can be selected, or a window which shows helpful information

5 Logic Layer Subsystems

The purpose of the Logic Layer will be to receive any requested information from the user or software and manipulate the return value into a usable data. The submitted data into this layer will undergo an analysis to ensure the request follows the guidelines presented in the UTA academic regulations.

5.1 Restriction Subsystem

The restrictions are put in place to ensure the actions and requested information by the user adhere to the rules specified by the UTA academic regulations. These restrictions will also take place in real-time to establish boundaries that will inhibit the user from making unacceptable choices.

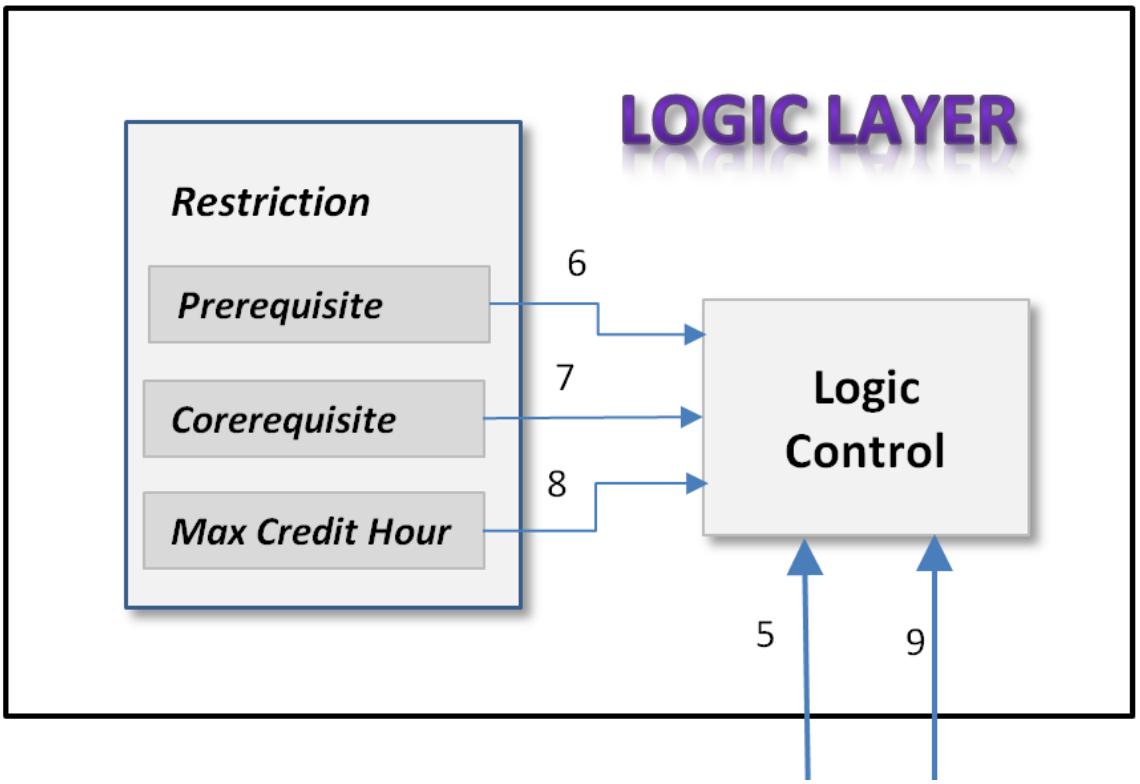

Figure 4: Logic layer and subsystems diagram

5.1.1 Assumptions

The layer will gain its restriction information and guidelines from the UTA academic regulations web page located at http://catalog.uta.edu/academicregulations.

5.1.2 Responsibilities

The restrictions will prevent the program from presenting information or allowing user choices that do not comply with the UTA academic regulations.

5.1.3 Subsystem Interfaces

The system will have a one-way interface with the Logic Control subsection. It will provide textual output information to the Logic Control for program decision making.

Table 7: Restriction Subsystem interfaces

ID	Description	Inputs	Outputs
#6	The subsection will send information towards the logic control and will manage the pre-requisites which are the courses required to be completed	The input will take form as a request for pre-requisite information on a specific course.	The output will take form as the pre-requisite course information on the requested course
#7	The subsection will send information towards the logic control and will manage the co-requisites which are the courses which can be taken alongside other courses	The input will take form as a request for co-requisite information on a specific course	The output will take form as the co-requisite course information on the requested course
#8	The subsection will send information towards the logic control and will manage the max amount of hours that a student can take during their semester	The input will take form as a request for the amount of hours for a specific course and the amount of hours that are on the user's current schedule	The output will take form as the hours on a specific course or the amount of hours the current schedule has

5.2 Logic control Subsystem

The Logic Control subsection will provide the GUI with the correct possibly manipulated data depending on whether the requested information or user input is acceptable under the UTA academic regulations.

5.2.1 Assumptions

The subsection will base its decisions, data manipulation, and mathematical equations based on the information and restrictions put in place by the Restrictions subsection.

5.2.2 Responsibilities

The main responsibility for the Logic Control will be to act as a mediator between the GUI and Data layer. Requested information from the GUI to the Data layer will undergo an analysis to ensure the guidelines are met.

5.2.3 Subsystem Interfaces

The subsection will contain a two-way interface between the GUI Control and Data Control subsections, and a one-way interface from the Restrictions subsection. Request information will be used as input and the requested data format as output.

Table 8: Logic Control Subsystem interfaces

ID	Description	Inputs	Outputs
#5	The subsystem will take the actions and events from the GUI layer and send the requested action to the Logic layer for approval or implementation	The input will be a request for an event/action to be taken in the Data layer or Logic layer	The output will be the requested data that was originally demanded from the GUI Control subsection or a confirmation message that an action was performed correctly
#9	The subsection will send information towards the Data Control located in the Data layer. The purpose is to communicate the Data Control subsection on what actions should be taken on the database, or what data is requested from there	The input will be data requested from the Data Control or a confirmation message that the actions/events requested by the GUI were performed	The output will be a request for a certain action/event to be taken in the database

6 Data Layer Subsystems

The data layer contains sections that provide simplified access to data stored in persistent storage. The layer also allows the saving and loading of any schedule if the users wishes to do so. Furthermore, the layer is also responsible for providing reliable data that will be analyzed in the logic layer and adjusted accordingly. Any more information needed at the logic control can also be requested to be passed from data layer. Once completed, it will be available for user display in terms of class schedule.

6.1 Database Subsystem

The subsystem allows the loading of course data and schedule into memory so that it can be accessed by the data control. All of the information will be populated by the system once the program starts or the executable is run the first time. The database comprises of different columns of data associated with classes. This includes course code, course title, course credit, course section number, current semester availability, course difficulty level, and course time and date. The difficulty level of each class will be based on a survey that will be conducted among senior Computer Science students in the Department of Computer Science and Engineering at UT-Arlington. The other information about the classes will be based on the fall 2018 semester degree plan for Computer Science students.

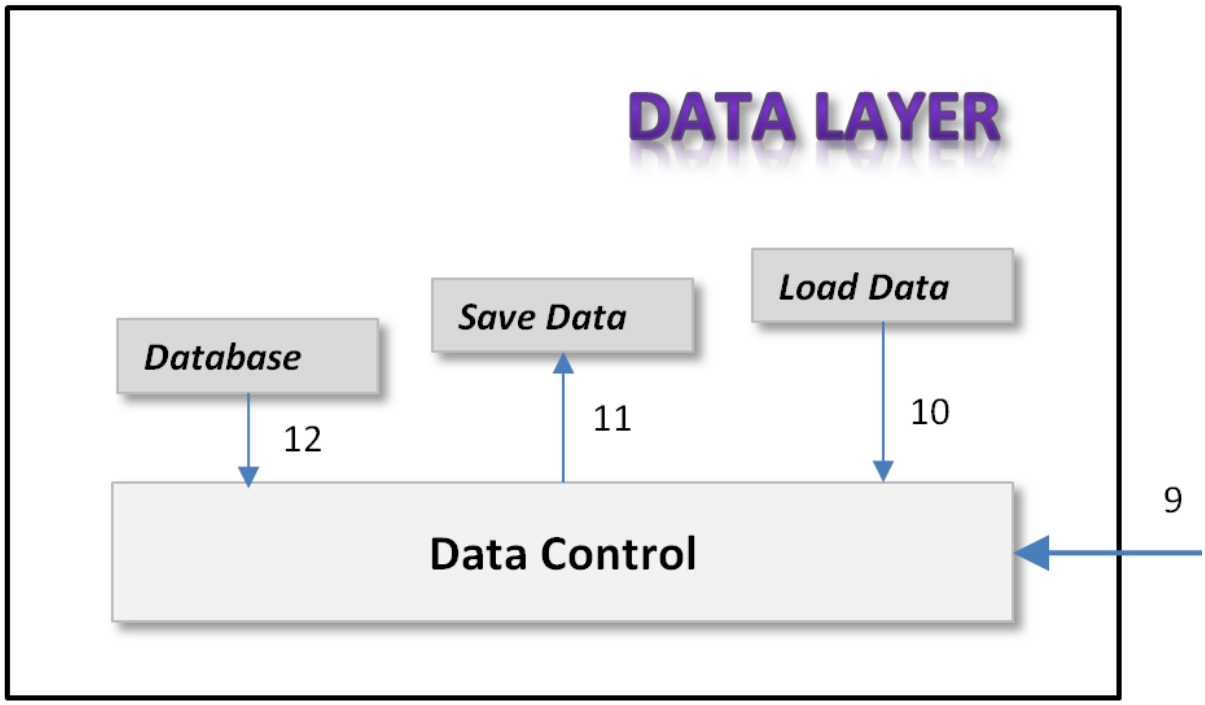

Figure 5: Data layer and subsystems diagram

6.1.1 Assumptions

The database contains consistent and accurate information that is not altered by any action. That is the data about each class remains populated once and can only be passed to the logic layer for processing and updates. There will not be any data modification (transformation, addition and removal). Additionally, in the case of a program crush the database stays unaffected and can be used to access information

again once the system is back.

6.1.2 Responsibilities

The responsibilities of the database system is store and organize data. It will be the basis for extracting, modifying, and searching for information about classes within a database.

6.1.3 Subsystem Interfaces

This subsystem will be a one way interface with the data control section. It will retrieve information related to outline course schedule.

Table 9: Database Subsystem interfaces

ID	Description	Inputs	Outputs
#12	The database will have a one-way interface with the data control section. It will provide course related output information to the control for appropriate information	N/A	The output will take a format that has information related to courses in the major.

6.2 Save Data Subsystem

The database is connected to the data control which handles the retrieval and retentions of data. This may involve passing the data to the logical layer for data query and processing before it is sent to the save data subsystem and formatted for saving preparation in terms of class schedule.

6.2.1 Assumptions

The manipulation of data in regards to course details such as co-requisites, prerequisites, and so on has already been processed and formatted by the logic layer to result in class schedules.

6.2.2 Responsibilities

The save data system provides relatively clear core information in regards to class schedule information for use so that students are aware of the selection of courses they will be taking during the upcoming semester.

6.2.3 Subsystem Interfaces

This subsystem will be a one way interface with the data control section. It will provide information related to courses within the major.

Table 10: Save Data Subsystem interfaces

ID	Description	Inputs	Outputs
#11	This section will retrieve course information about classes that has been arranged into a class schedule.	The input will take a format data that has filtered information related to courses in the major.	The output for this section includes an outlined list of classes for students to take by taking into consideration details about each course work from the data control

6.3 Load Data Subsystem

Unlike the save data subsystem, the load functionality allows class schedules created initially by users to be passed to the data control if the students want to make further modifications to their preexisting schedule. Once the information is passed to the data control, It will be further processed at the logic layer based on the changes requested and the degree plan requirements for undergraduate computer science students in the Department of Computer Science and Engineering at UT-Arlington.

6.3.1 Assumptions

The load data system provides an information that is valid and was accurately generated by the software.

6.3.2 Responsibilities

The load data system delivers data so that any further changes or important modifications to class schedule can be implemented.

6.3.3 Subsystem Interfaces

This section will contain a one way interface with the data control section. It will provide formatted class schedule information to the data control for examination before they are passed to the logic layer for manipulation.

Table 11: Load data Subsystem interfaces

ID	Description	Inputs	Outputs
#10	This section will insert course information about classes that has been arranged into a class schedule	The input for this section comes from retrieving previously produced class schedule that made use of data related to courses in the major	The output will take a format that has filtered information related to current class schedule so that it can be further manipulated and adjusted

6.4 Data Control Subsystem

The data control subsystem will provide the logic layer with accurate data for processing and manipulation. On other hand, it will also route user scheduling data from/to its subsystem for loading and saving.

6.4.1 Assumptions

The control will forward information for processing as raw data that was stored in place about courses related to Computer Science Major.

6.4.2 Responsibilities

The main responsibility of the Data Control is to provide data flow to the logic when information is requested, and receive scheduling data when saving is asked for by the user.

6.4.3 Subsystem Interfaces

This section will contain a bidirectional interface to the Logic control section. Any previous information that is loaded will be used as an input while requested information will be used as the output.

Table 12: Data Control Subsystem interfaces

ID	Description	Inputs	Outputs
#9	This section will send information towards the Logic control in the Logic layer. The purposed is to pass data to it so that actions can be taken on the data in logic layer	The input for this section comes as requests from the logic control	The output will be the filtered information related to current classes within the major

References