(CS 364-02)

(Homework Assignment Tracking Application)

(Final Version)

Software Design Document

This Page Intentionally Left Blank

1. INTRODUCTION

1.1. Purpose

1.1.1. SDD Document

This document provides a detailed description of the software architecture, systems, and subsystems required to complete the Homework Assignment Tracking Application project. The document shall provide a design description for the system on both the client and server side, as well as document network synchronization. The document shall also describe how user interface elements will be implemented. This document conforms to the Software Design Document (SDD) Template from Construx Software Builders, Inc. [1] and the Software Design Document (SDD) Template from the Concordia University-Montreal Department of Computer Science [2].

1.1.2. Homework Assignment Tracking Application (HATA)

The purpose of the Homework Assignment Tracking Application is to provide a dynamic, organized, and efficient method for students to view all of their upcoming assignments. This application is specifically designed for students during their educational career. HATA shall help students decrease the uncertainty of when an assignment is due, and improve student productivity by reducing overall frustration among student users by providing a single location for students to access course tasks.

1.2. Scope

HATA shall be a mobile phone application that shall be built on both Apple's iOS and Google's Android platforms. HATA shall also integrate with the I-Learn system. In addition, HATA shall have a web service for storing and handling created data that shall exist outside of the I-Learn system. HATA shall also integrate with third party user authentication systems.

HATA is designed to allow students to directly access their course assignment information in a list or calendar view. The assignment descriptions in the list or calendar shall pull the assignment information directly from I-Learn so the student can see exactly what the instructors expect the student to submit and when the assignment is to be submitted. Notifications from the student's mobile device shall also remind the student when assignments are due. Students shall be able to see assignment changes within minutes whenever their instructors make changes to an assignment.

The objective of HATA shall be to create, manage, and edit assignments. This shall benefit students by giving the students tools to plan and manage the student's

available time and assignment workload. As students mark assignments as complete, students shall benefit from increased efficiency through a smaller mental model as these assignments are removed from their assignment list.

In order to facilitate ease of use for HATA, there shall be search and filter functions which allow the student to quickly find which assignments are due soon, or find the assignments that are assigned during any given week for any given course. Through the use of tags and color coding, the student shall be able to easily discern which assignments are due and when they are due with just a glance at the application screen.

HATA shall also provide the ability for the software to be used offline without an internet connection. This shall benefit students by giving the students the convenience of seeing assignments in geographic locations where internet is limited or not available.

The last high level objective for HATA is to notify students of upcoming due dates for assignments. This shall benefit students as students shall receive reminders to help the students avoid turning in assignments late.

1.3. Limitations

We are using Google Docs as our submission and are not submitting this document in Microsoft Word. There are some minor formatting challenges associated with using Google Docs. Most notably:

- large spaces between images used as diagrams,
- spaces between images used as diagrams and their labels/descriptions,
- inability to properly link between sections on the same page,
- limited control over both horizontal and vertical spacing and alignment due to the lack of precise ruler control,
- intentionally irregular spacing in section 4 so as not to disrupt cohesive subsections,
- limited control over the styling of the Table of Contents,
- limited control over the styling of headings,
- and limited font variation options for document sections.

This document will only have up to 5 components of descriptions (eg. X.x.x.x.x).

Due to time constraints and personnel inexperience the following items were not included in this document:

- UML diagrams of subordinates of the major components in section 4.
- Any subordinate, algorithm, or data breakdown further than 4 components deep in parts of section 4 and 5 levels deep in other areas of section 4.

- Any architecture diagrams other than structure charts and flow diagrams that are typically used to represent a system. Diagrams such as hardware diagrams, network diagrams, state transition diagrams, process flow diagrams, and entity relationship diagrams for the mobile client were not included.
- A comprehensive data architecture and data dictionary for section 3. A limited ERD is included for the HATAWebAPI, but that is the extent of the data architecture.
- User Interface designs for all of the stakeholder scenarios and interfaces of the HATA Mobile Application. Only a limited subset involving saving a task and viewing the calendar is included.
- Consistent header and component identifiers in section 4. Most component headers in section 4 match their component identifier. However, some inconsistencies were left in the document due to the complication of updating the diagrams in Appendix 7 that had the inconsistent names.
- A design specification for the HATAWebAPI Settings Controller due to the lack of contributions of the subject matter expert.
- In Appendix 7, due to time constraints and lack of personnel experience, Diagrams 18 (7.18), 19 (7.19), and 20 (7.20) have been removed, but the place markers have been kept to keep the document consistent with its references.
- Requirement 3.2.3.1 from the SRS has been altered in section 5.3 of the SDD for usability purposes.

Due to use of Google Docs, subordinate numbering levels were adjusted to demonstrate parent/child relationships and do not follow normal numbering standards (i.e. 4.2 content should fall within section 4.1.41). This "flattening" allows for needed detail in the application structure to demonstrate SDD knowledge and understanding.

An intentional Blank Page was added on page 2 due to Google Doc limitations. This caused numbering issue with entire SDD.

Due to time constraints and limited resources, the following Software Requirement Specification [3] requirements were not addressed in this Software Design Document:

- 3.2.2.4
- 3.2.3.2
- 3.2.5.2
- 3.2.5.3
- 3.2.6.8
- 3.2.6.9

- 3.2.7.1
- 3.2.8.2
- 3.2.9.1
- 3.2.10
- 3.2.10.1
- 3.2.10.2
- 3.2.10.3
- 3.2.10.4
- 3.2.11.2
- 3.2.11.3
- 3.2.11.4
- 3.2.11.11
- 3.2.12.2
- 3.2.12.3
- 3.3.2.7
- 3.3.2.9
- 3.3.2.10
- 3.3.2.11
- 3.3.3.1
- 3.3.3.5
- 3.3.4.5
- 3.3.4.8
- 3.3.4.10
- 3.3.6.1
- 3.3.6.2
- 3.3.6.6
- 3.3.6.8
- 3.3.6.10
- 3.3.6.11
- 3.3.6.13
- 3.4.1.1
- 3.5.1.2
- 3.5.1.3
- 3.6
- 3.7.2.1
- 3.8.2

Due to time constraints and limited resources, the following Software Requirement Specification [3] requirements, that were requested but not required, were not addressed in this Software Design Document:

- 3.2.2.5
- 3.2.3.7
- 3.2.4.2
- 3.2.6.3
- 3.3.6.14

1.4. Reference Material

[1] Construx Software Builders. "Software Design Document (SDD) Template.". [Online]. Available: https://sovannarith.files.wordpress.com/2012/07/**sdd_template**.pdf

[2] "Software Design Document (SDD) Template,". [Online]. Available: http://docplayer.net/8184133-software-design-document-sdd-template.html.

[3] CS364-02. "Software Requirements Specification". [Online]. Available:

https://docs.google.com/document/d/1bsoLJ8OgCPiZQpjjINfAKQzDmbezUiPZy3f1pTAUFlE/edit#

[4] "Node.js MVC Example Implementation with Xiao Blog Engine," Xiao Blog: A Node.js blog engine, simple, fast, & free. [Online]. Available: http://xiaoblog.digibrady.com/features/nodejs-mvc-example-implementation-with-xiao-blog-engine.

1.5. Definitions

1.5.1. Airplane Mode - A mobile device setting that suspends radio-frequency signal transmission by the device, thereby disabling Bluetooth, GPS, telephony, and Wi-Fi once the setting is activated.

1.5.2. App Data - The information that the mobile app uses to provide its intended functionality. This includes information such as assignments, student-provided information, courses, business logic constraints, pictures, and authorization tokens.

1.5.3. Approve - When the student gives permission for the given operation or setting, usually through the form of a window asking for confirmation.

1.5.4. Archived - The state of a task when its information has been stored in the internal storage of the mobile device.

1.5.5. Assignment - An objective that a student needs to accomplish. An assignment can be created by an instructor on I-Learn. An assignment is a task when the assignment is created by a student on HATA. An assignment

can be assigned by an instructor to one or more students who are part of a course. The assignment has the following attributes: assignment name, assignment course, assignment due date, assignment notes, assignment link, and an assignment reminder setting. The assignment may also contain multiple assignment reminders, assignment difficulty, assignment start date, assignment priority, and assignment completion percentage. If an assignment was created by an instructor, a student can only modify an assignment's notes or assignment reminder settings.

1.5.6. Assignment Completion Percentage - The metric used to judge how much progress has been made on a assignment.

1.5.7. Assignment Difficulty - The metric set by the user to rank the amount of time and effort a given assignment shall require.

1.5.8. Assignment Information - The attributes of an assignment including but not limited to the assignment name, course, due date, and assignment notes.

1.5.9. Assignment List - The default list of tasks displayed upon entry to HATA that lists tasks by the set filter and their general information.

1.5.10. Assignment List Width - The set width of the Assignment List allowed to display tasks.

1.5.11. Available Network Communication - When either a network packet has been received within the last 500ms or the mobile device OS has notified the application that network communication is available.

1.5.12. Backend Web Service - An application programming interface hosted by one or more servers that user facing applications or devices can make calls to by sending and receiving data through a series of HTTP requests.

1.5.13. Block - To disable the ability to perform an action.

1.5.14. Character Limit - The maximum number of letters, numbers, symbols, and spaces.

1.5.15. Color-Coded - The assigning of colors to indicate a specific commonality amongst items.

1.5.16. Completed Assignments List - A list of assignments the user has marked as completed.

1.5.17. Consistent Internet Connection - Uninterrupted communication between the data source and HATA.

1.5.18. Course - A**n** organizational unit that is focused on the teaching of a specific subject matter with a beginning and end time that is comprised of one or more instructors and one or more students.

1.5.19. Create a New Task - Entering details for a task that does not exist and that are then saved and displayed on the assignment list.

1.5.20. Daily Calendar View - A view that allows the user to only see assignments and tasks that correspond that the selected day.

1.5.21. Date Selector - Controls provided by the operating system that allow the user to pick a date.

1.5.22. Default Input Method - The means by which the user provides the requested information to HATA.

1.5.23. Default Tag Options - A predetermined label attached to Quizzes, Discussion Board, Reading, Group Project, Activity, Individual Project, Course Name, Due Dates, and Contact Information to aid the user in the task of filtering.

1.5.24. Default View - The characteristics and method of viewing a given part of HATA that shall be set prior to modifications of the user.

1.5.25. Delete Time - The time that is set in the application for how long a assignment shall remain on the assignment list after the assignment due date has passed.

1.5.26. Deleted Task - A task that has reached its due time as well as its delete time and has been removed from HATA.

1.5.27. Device Notification - A message that appears in the notifications window on the mobile device alerting the user to an update of some kind.

1.5.28. Device System - The internal operations and settings of the used mobile device.

1.5.29. Due Date - The date (month and day) and time (hours, minutes, seconds) that an assignment or task must be completed.

1.5.30. Easy Access Mode - When enabled, this allows the user to access their information by opening the application without requiring the user to provide authenticate.

1.5.31. Error Message - A window that appears over the regular application alerting the user to failure to complete an operation.

1.5.32. **Express.JS** - A web application framework designed for building web applications and APIs.

1.5.33. **Filtering -** A process that sorts a set of objects based on the labels attached to the objects.

1.5.34. **General Task Delete Time** - The delete time that shall be the default configuration when a task is created.

1.5.35. **Grade Weight** - The percentage the assignment is worth within the overall total grade of the course.

1.5.36. **Group Project** - A task that has been assigned to be completed with effort from multiple students.

1.5.37. **Home Page -** The default view of the application, from which the user may navigate to any other view.

1.5.38. **Hypertext Transfer Protocol** - A method of packaging and transmitting data through the use of network communication.

1.5.39. **I-Learn** - The third and most current version of the learning management system that the university BYU-Idaho uses for the delivery of courses to students.

1.5.40. **I-Learn Synchronization** - The process of retrieving assignment information that is new or has changed since the last network communication with I-Learn.

1.5.41. **Immediate Deletion** - As soon as a task reaches its due date, the task is removed from HATA.

1.5.42. **Incomplete assignment** - An assignment that a student has not marked as completed.

1.5.43. **Inhibit** - To make difficult or impossible.

1.5.44. **Input** - Putting data into HATA.

1.5.45. **Instructor** - The individual that oversees the instruction of a course as well as creates assignments in I-Learn.

1.5.46. **Learning Management System** - A software application that automates the administration, tracking, and reporting of training events or courses.

1.5.47. **Low on Storage** - The state of the mobile device when the mobile device lacks the necessary available memory to complete a basic operation.

1.5.48. **Mobile App** - An application developed for and run on a mobile device.

1.5.49. **Mobile Data** - The communication mechanism where information is both sent and received through the cell phone communication transmitter on a mobile device.

1.5.50. **Model-View-Controller** - A software design pattern that manages complexity and provides a separation of concerns. Software is organized into areas pertaining to data and business logic (Model), presentation of data and logic, as well as receiving input from the consumer of the application (View), and the code that arranges data and communication between both the model and the view (Controller).

1.5.51. **Network Communication** - Communication between two or more devices where information is both sent and received through a network interface such as a cell phone communication transmitter on a device or through Wi-Fi or Ethernet.

1.5.52. **Network Problems** - An error or errors in communication between a source and HATA.

1.5.53. **Notification -** A message delivered from a software application to a mobile device without a prompt from the user.

1.5.54. **Notification Time** - The time that is set in the application for how long before the assignment is due that a notification should appear.

1.5.55. **Notify** - Alerting the user to the progress, completion, or failure of an operation.

1.5.56. **Object Class** - A grouping of related methods and data attributes from which objects in a software system are instantiated from.

1.5.57. **Offline** - The state of the mobile device when the mobile device has no data connection.

1.5.58. **Onscreen Keyboard** - A virtual keyboard displayed on a touch device.

1.5.59. **Ordered Location** - The position of a task in the assignment list set according to the student-specified filter.

1.5.60. **ORM** - An object-relational mapping which allows for the mapping of database tables to objects usable in an object-oriented programming language.

1.5.61. **Password** - A unique security phrase a student shall use to connect HATA to the student's I-Learn account.

1.5.62. PhoneGap - A software development framework by Adobe System, which is used to develop mobile applications.

1.5.63. PIN Code - A password a student shall use to authenticate their login credentials when logging into HATA after they have connected HATA to their I-Learn account, or when altering account settings.

1.5.64. Profile Picture - An image set up by the student for display in HATA.

1.5.65. Progress Bar - A graphical bar that shows the progress of I-Learn synchronization.

1.5.66. QA Engineer - A type of user that verifies the functionality of HATA is correct.

1.5.67. Refresh of Content - When the application calls out to I-Learn to fetch data and updates that data stored on the user's mobile device.

1.5.68. Remember - When the application stores a user's selection in the mobile device's internal memory as well as on the cloud so as to keep settings and information consistent between uses.

1.5.69. Required Text Field - A text field that must contain information before the task can be saved.

1.5.70. REST URL - A collection of URLS, many of which shall have relational links.

1.5.71. Restore - To retrieve a task and make the task appear as though the task had not been deleted.

1.5.72. Saving a Task - When a task has been created or modified and the user selects to store the creation or modification on the mobile device's storage and the cloud.

1.5.73. Select - When an item is selected, the item is "touched" by the user.

1.5.74. Semester - A 14-week period at BYU-Idaho during which a student can enroll in one or more courses.

1.5.75. Server - A computer program or a device that provides functionality for other programs or devices called "clients". See 2.4.

1.5.76. Setting - A configuration for the application.

1.5.77. Settings Window - The window that contains all of the general settings that can be modified by the user for the entire application.

1.5.78. Shall - The app is required to or must have this functionality.

1.5.79. **Should** - The app is expected to have this functionality, but the functionality is not a requirement.

1.5.80. **Slider** - A switch-like UI element that is toggled by the user to an "on" or "off" state.

1.5.81. **Software Method** - An atomic piece of functionality that belongs to an object class in a software system.

1.5.82. **Sort/Sortable** - The ability of a given list of tasks to be ordered by a user specified metric.

1.5.83. **Squelize** - An ORM for Node.JS that supports MySQL.

1.5.84. **SSL Certificate** - A Secure Socket Layer Certificate, which is a file that is used by servers participating in network communication to encrypt and validate data being transmitted using the SSL protocol.

1.5.85. **SSL Vulnerability Scanner** - A software system that scans servers to verify that the **server** is using SSL properly and is not exposed to any known security holes or vulnerabilities. This can be a stand-alone software or a web service.

1.5.86. **String -** An object that represents a sequence of characters, usually plaintext.

1.5.87. **Student** - This is a type of user of the system who participates in a course and receives assignments that must be completed as part of being a member of that course.

1.5.88. **Synchronize** - Connecting to the external source and verifying that the shared information is consistent between the two and resolving differences if differences exist.

1.5.89. **Tags -** A label attached to assignments to aid the user in the task of filtering.

1.5.90. **Task -** A task is a type of assignment that has been created by a student on HATA.

1.5.91. **Task Course** - The course that a student has associated a task with.

1.5.92. **Task Creation Screen** - The window where the user inputs the information about the task he or she is creating.

1.5.93. **Task Details** - All of the detailed information that has been provided for a task.

1.5.94. **Task Details Screen** - The window displayed to a student on HATA that shows all of the attributes of a task.

1.5.95. Task Link - A URL that connects to a website which has additional information to read or learn about a task.

1.5.96. Task Name - The short, student or I-Learn provided, description of what a task is.

1.5.97. Task Notes - Additional textual information that the student has associated with a task that further describes the task or is information that the student wants to remember about the task.

1.5.98. Task Priority - The level of importance set by the user to indicate the importance given to a specific task.

1.5.99. Task Reminder Setting - An attribute of a task that is the time from the task due date, that a student shall be sent a notification that the task is due. The valid values of this attribute are: 10 minutes, 30 minutes, 1 hour, 2 hours, 4 hours, 12 hours, 1 day, 2 days, 3 days, 5 days, and 1 week.

1.5.100. Task Start Date - The date at which the user may begin work on a task.

1.5.101. Team Members - The fellow members of a course who have chosen or been assigned to participate with the student in work on a task.

1.5.102. Timed Out - When an operation has not completed in its allotted time and the operation is cancelled.

1.5.103. Unit Test - A unit is the smallest testable part of an application like functions, object classes, procedures, interfaces. A unit test verifies that the unit under test fulfills all of its required functionality.

1.5.104. User Authentication System - A third party authentication system that authorizes a user and provides a token to the calling application to use for current and future identification of the user.

1.5.105. User Image - A generic picture used as a placeholder for a profile picture until said profile picture is configured by the student.

1.5.106. Username - An identifying title unique to a specific user.

1.5.107. Verified Defect - A problem identified by a stakeholder in the application software that has been independently verified by a software engineer or quality assurance engineer.

1.5.108. Verify - To check a certain item against a known value or state.

1.5.109. Voice Input - The process of translating speech to text which can then be used in a text input field.

1.5.110. Wi-Fi - A technology that allows a device to send and receive information through a wireless local area network (WLAN).

1.6. Acronyms and Abbreviations

1.6.1. BYU-Idaho - Brigham Young University - Idaho

1.6.2. GPS - Global Positioning System

1.6.3. HATA - Homework Assignment Tracking Application

1.6.4. HTTP - Hypertext Transfer Protocol

1.6.5. HTTPS - HyperText Transfer Protocol Secure

1.6.6. IEEE - Institute for Electrical and Electronic Engineers

1.6.7. iOS - iPhone Operating System

1.6.8. LMS - Learning Management System. See section 1.4 for definition.

1.6.9. OS - Operating System

1.6.10. PIN - Personal Identification Number

1.6.11. QA - Quality Assurance

1.6.12. SMS - Short Message Service

1.6.13. SSL - Secure Socket Layer

1.6.14. Sync - Synchronize

1.6.15. Synced - Synchronized

1.6.16. UI - User Interface

1.6.17. URL - Uniform Resource Locator

1.6.18. Wi-Fi - Wireless Fidelity

1.6.19. WWW - World Wide Web

1.7. Non-Conformance

This document has sections that do not conform with the SDD Templates [1, 2]. Section 2 System Overview of the SDD Template has been removed as section 1.2 Scope address the System Overview. Section 1.3 Document Overview of the SDD Template has been removed as Section 1.1.1 and the Table of Contents of this document fulfill the purpose of a Document Overview. This results in Section 3 through 8 being renamed as Section 2 through 7 respectively.

Section 5 has been renamed from Human Interface Design to User Interface Design. Section 5.3 was dropped as the objects and actions are included under each screen image that they belong to in Section 5.2.

2. SYSTEM ARCHITECTURE

2.1. Architectural Design

The system is divided into two sets of client-server architecture components. The first client-server consists of the Mobile HATA System as the client, and the HATAWebAPI as the server (Appendix 7.1 Diagram 1).

The second client-server relationship consists of the HATAWebAPI system as the client, and I-Learn's web API system as the server. Please see the Architectural Decisions section 2.3.1 for more information.

Each high level system and subsystem is described in this section with its purpose, function, and the subordinate components. Diagrams for each component are referenced in the component title and can be found in Section 7.

2.2. Decomposition Description

2.2.1. Mobile HATA System

The Mobile HATA System (Appendix 7.1 Diagram 1) is responsible for tracking and managing assignments and tasks for students on their mobile device. HATA Mobile Application is built using PhoneGap to cross compile the codebase to work on both Android (in Java) and iOS devices (using Objective-C). The mobile application is built following the OOP paradigm. The languages used are Javascript and HTML5, with device platform specific code created as needed to handle components that PhoneGap is not capable of providing (see Architectural Decisions Section 2.3.1 for the rationale behind this). For more information see Section 4.1 for the component design of this system.

2.2.1.1. Client Synchronization:

The Client Synchronization subsystem is responsible for all of the mobile device data storage and communication with the HATAWebAPI system. Data Storage information can be seen in section 3 and the component definition of this subsystem is viewable in section 4.2. In Diagram 2 (Appendix 7.2) you can see the structure chart breakdown of the four components of this system: Internal Storage, Mobile Network Communication, Server Network Communication, and Cloud Storage. The UML Component diagram in Diagram 3 (Appendix 7.3) shows the available interfaces and how they are used in the ClientSynchronization subsystem.

2.2.1.2. AppGUI

The AppGUI subsystem is responsible for the entire presentation of the Mobile HATA System. From the structure chart in Diagram 4 (Appendix 7.4) you can see that AppGUI is broken down into four main areas of the app: the Home Page, Assignment View, Calendar View, Task Creation View, and Search Results View. A more detailed class diagram of this subsystem can be found in Diagram 6 (Appendix 7.6). The component breakdown of this subsystem can be seen in section 4.4.

The Home Page component is the main display screen (Appendix 7.22) for the mobile application. From the home page the student can navigate to several other areas of the app. These other sections, which are reachable from the home page, are displayed in Diagram 8 (Appendix 7.8). The default display for the Home Page will be the Calendar View.

When a user performs a search on the Home Page, AppGUI will load the Search Display Results Screen as seen in Diagram 9 (Appendix 7.9)

The Calendar View of AppGUI shows the different assignments and tasks the student has across several different viewing formats as seen in Diagram 10 (Appendix 7.10). The Calendar view can also be organized according to day, week, and month.

The Task Creation Screen is shown in Diagram 11 (Appendix 7.11). The task creation screen is where the user can generate a task by inputting a task name and selecting a course, date time, and reminders, with the option of entering notes or tags.

2.2.1.3. Account Management

The Account Management subsystem is responsible for identification of user account information and management of user account information. A breakdown of its classes can be seen in Diagram 5 (Appendix 7.4). Its component breakdown is found in section 4.3.

2.2.1.4. Options

The Options subsystem handles the management of all the settings in the Mobile HATA System. The Options menu can be accessible from AppGUI. The different settings areas reachable from the Options menu can be seen in Diagram 12 (Appendix 7.12). Diagram 13 (Appendix 7.13) shows a class breakdown of the Option subsystem. The component description can be found in section 4.5.

2.2.2. HATAWebAPI System

The HATAWebAPI system handles the synchronization between I-Learn and Mobile HATA System and the synchronization of data between the HATA Mobile Application clients and the cloud storage of the HATAWebAPI System.

HATAWebAPI uses the Express.JS framework. A Flow Diagram depicting how the Express.JS framework works in our system can be seen in Diagram 14 (See Appendix 7.14). The HATAWebAPI fits into the Express.JS framework by setting up REST URLs that are connected to JavaScript controllers to accomplish the saving, creating, and persisting of HATA data. HATAWebAPI also communicates to the I-Learn web API system to pull assignment information. A breakdown of the classes that sit on top of the Express.JS framework in the HATAWebAPI can be seen in Diagram 15 (See Appendix 7.15). The component description of this system can be found in section 4.6.

2.3. Design Rationale

2.3.1. Client Server Architecture

HATA Client communicates to HATAWebAPI to get a list of assignments and tasks. HATAWebAPI talks back and forth to I-Learn to receive assignment lists (Appendix 7.16 Diagram 16). Merging is done on the HATAWebAPI. This Client Architecture was selected to allow us to create our own cloud and to keep the cloud synced with bth I-Learn and the Hata Mobile Application.

2.3.2. Client Development

Client application has a single code base that is processed by PhoneGap (Appendix 7.17 Diagram 17). This processing yields the artifacts necessary for deployment on the Android and IOS platforms. This development platform was chosen due to a small team and simple design. PhoneGap allows us to create the application in a reasonable amount of time.

3. DATA DESIGN

3.1. Data Description

The data for HATA is stored as table rows in relational databases. The data for the HATA Mobile Application is stored in a SQLLite database on the mobile device of the student. Any changes of data on the mobile device are synchronized with the data stored on the HATAWebAPI. The stored data for the HATAWebAPI is stored in a MySQL server cluster.

The HATAWebAPI's main data entities are Assignment, Course, AuthControllerData, and TaskControllerData. A Entity Relationship Diagram of these entities can be found in Diagram 21 (See Appendix 7.21 Diagram 21).

3.2. Data Dictionary

Name	Description	Type
Assignment	Contains information relating to a single assignment (name, due date, date assigned, affiliated course).	Entity
Course	Contains information about a specific course (name, course code, teacher, credits).	Entity
AuthControllerData	Contains authentication information related to the user (username, email, password).	Entity
TaskControllerData	Contains information about tasks for the user the do.	Entity

4. COMPONENT DESIGN

4.1. HATA Mobile Application

4.1.1. Component Identifier:

HATAMobileApplication

4.1.2. Purpose:

This component shall handle all information that HATA displays to the user.

This component shall handle the creation of assignments and tasks for HATA on a mobile device.

This component shall track assignments and tasks for HATA on a mobile device.

4.1.3. Function:

This subsystem is broken down into five main components. It uses the AppGUI component to handle all of the HATA display on the mobile device. The creation of tasks are handled by the Task Creation Screen component, and the tracking of assignments and tasks for HATA on a mobile device are done by the ClientSynchronization component. Device settings are handled by the Options sub-system, and the creation of new tasks in this system are handled by the Task Creation Screen subsystem.

4.1.4. Subordinates:

4.1.4.1. ClientSynchronization

4.1.4.2. Account Management

4.1.4.3. AppGUI

4.1.4.4. Options

4.1.5. Dependencies:

4.1.5.1. Requires installation on a mobile device.

4.1.5.2. Needs user data.

4.1.5.3. Needs user input.

4.1.6. Interfaces:

4.1.6.1. None

4.1.7. Data:

4.1.7.1. Data from the user, class, teacher, and school; assignment and task information; information from the device, including date and time

4.2. ClientSynchronization

4.2.1. Component Identifier:

ClientSynchronization

4.2.2. Purpose:

This component shall manage the communication and synchronization of the HATA mobile application with the HATAWebAPI.

4.2.3. Function:

Shall manage client side communication with the HATAWebAPI.

Shall communicate and synchronize data with HATAWebAPI every five minutes.

Shall overwrite changes in InternalStorageController to assignment data that has been updated and set to overwrite by HATAWebAPI.

Shall send multiple requests for data when the HATAWebAPI sends a flag saying the request payload exceeds 512KB of data.

Shall notify AppGUI that the synchronization request has timed out if the request exceeds longer than one minute.

4.2.4. Subordinates:

4.2.4.1. InternalStorageController:

The location where all app settings and information shall be stored on the mobile device.

4.2.4.2. CloudStorageController:

The location where all app settings and information shall be stored on the cloud.

4.2.4.3. ServiceNetworkCommunicationController:

This shall pass information with the web service securely.

4.2.4.4. MobileNetworkCommunicationController:

The network communication that shall be used to synchronize with the web service.

4.2.5. Dependencies:

4.2.5.1. Requires communication from the HATA mobile application and the HATAWebAPI.

4.2.5.2. Requires Cloud to store settings and information with the CloudStorageController.

4.2.5.3. Requires mobile device to store settings and information.

4.2.6. Interfaces:

4.2.6.1. None

4.2.7. Data:

4.2.7.1. Communication data from the HATA mobile application; HATAWebAPI; settings from the user; mobile device data including storage settings

4.2.8. InternalStorageController

4.2.8.1. Component Identifier:

InternalStorage

4.2.8.2. Purpose:

This component shall store and remove data.

4.2.8.3. Function:

The location where all app settings and information shall be stored on the mobile device.

4.2.8.4. Subordinates:

4.2.8.4.1. **AppData**:

The information that the mobile app shall use to provide its intended functionality.

4.2.8.4.2. **NetworkCommunicationStatus**:

The availability of HATA to communicate with the mobile network. If HATA can communicate, the status shall be available. If HATA cannot communicate, the status shall be unavailable.

4.2.8.5. Dependencies:

4.2.8.5.1. None

4.2.8.6. Interfaces:

4.2.8.6.1. None

4.2.8.7. Data:

4.2.8.7.1. AppDataModel

4.2.9. AppDataModel

4.2.9.1. Component Identifier:

AppData

4.2.9.2. Purpose:

This component shall be the way HATA will store the settings.

4.2.9.3. Function:

The information that the mobile app shall use to provide its intended functionality.

4.2.9.4. Subordinates:

4.2.9.4.1. UserSettings:

The configuration for the application that shall be set upon download and modifiable by the user.

4.2.9.4.2. Assignment:

An objective that the user needs to accomplish that shall be pre-populated by a synchronization with I-Learn.

4.2.9.4.3. Tasks:

A task is a type of assignment that shall be created by a student on HATA.

4.2.9.5. Dependencies:

4.2.9.5.1. None

4.2.9.6. Interfaces:

4.2.9.6.1. None

4.2.9.7. Data:

4.2.9.7.1. UserSettingsModel; AssignmentsModel; TasksModel

4.2.10. UserSettingsModel

4.2.10.1. Component Identifier:

UserSettings

4.2.10.2. Purpose:

This component shall provide the user with a place to view settings.

This component shall allow for modification of the user's application-wide settings.

4.2.10.3. Function:

The configuration for the application that shall be set upon download and shall be modifiable by the user.

4.2.10.4. Subordinates:

4.2.10.4.1. TaskReminderSetting:

An attribute of a task that is the time from the task due date that a student shall be sent a notification that the task is due.

4.2.10.4.2. MuteSetting:

A user provided time period during which notifications shall not be sent.

4.2.10.4.3. DeleteTimeSetting:

The time that is set in the application for how long a assignment shall remain on the assignment list after the assignment due date has passed.

4.2.10.4.4. ArchiveDeletedSetting:

Whether or not a task's or assignment's information shall be stored in the internal storage of the mobile device when it has been deleted.

4.2.10.4.5. LowStorageDeletionSetting:

Whether or not a task or assignment's archived information shall be removed when the mobile device is low on storage.

4.2.10.4.6. SMSNotificationSetting:

Whether task or assignment notifications shall be delivered as a mobile notification or a SMS notification.

4.2.10.4.7. ProfilePictureSetting:

The user provided image provided that shall represent his or her account.

4.2.10.4.8. ContactInformationSetting:

The information the user shall provide, including his or her name and the communication method that shall be used to contact the user.

4.2.10.4.9. UsernameSetting:

The specific string that shall identify the user that is unique from all other users.

4.2.10.4.10. SocialMediaSetting:

Information that shall connect HATA to I-Learn, Google+, and Facebook.

4.2.10.4.11. BackgroundRefreshSetting:

Whether or not HATA shall fetch updated assignments in the background, including new or changed assignments from instructors or messages from other students.

4.2.10.4.12. DailyOverviewSetting:

Whether or not a notification shall be sent at the beginning of each day with a summary of the assignments or tasks due that day.

4.2.10.4.13. MobileDataUsageSetting:

Whether or not HATA shall use a mobile data connection for synchronizing in the absence of Wi-Fi connectivity.

4.2.10.5. Dependencies:

4.2.10.5.1. None

4.2.10.6. Interfaces:

4.2.10.6.1. The user settings window shall serve as the interface for the UserSettingsModel.

4.2.10.7. Data:

4.2.10.7.1. TaskReminderSettingModel; MuteSettingModel; DeleteTimeSettingModel; ArchiveDeletedSettingModel; LowStorageDeletionSettingModel;

SMSNotificationSettingModel; ProfilePictureSettingModel; ContactInformationSettingModel; UsernameSettingModel; SocialMediaSettingModel; BackgroundRefreshSettingModel; DailyOverviewSettingModel; MobileDataUsageSettingModel

4.2.11. AssignmentModel

4.2.11.1. Component Identifier:

Assignment

4.2.11.2. Purpose:

This component shall store the provided information about an assignment.

4.2.11.3. Function:

An objective that the user needs to accomplish that shall be pre-populated by a synchronization with I-Learn.

4.2.11.4. Subordinates:

4.2.11.4.1. Name:

The title that shall be assigned to an assignment or task.

4.2.11.4.2. Course:

The class or course that shall be associated with a specific assignment or task.

4.2.11.4.3. Notes:

The notes entered by the user that shall pertain to a specific assignment or task.

4.2.11.4.4. DueDate:

The date (month and day) and time (hours, minutes, seconds) that an assignment or task shall be completed.

4.2.11.4.5. ReminderTime:

An attribute of a task or assignment that is the time from the task due date, that a student shall be sent a notification that the task is due.

4.2.11.4.6. Status:

Whether or not an assignment or task is complete.

4.2.11.5. Dependencies:

4.2.11.5.1. None

4.2.11.6. Interfaces:

4.2.11.6.1. None

4.2.11.7. Data:

4.2.11.7.1. Assignment name; course; notes; due date; reminder time; status of an assignment

4.2.12. TaskModel

4.2.12.1. Component Identifier:

Task

4.2.12.2. Purpose:

This component shall store the provided information about a task.

4.2.12.3. Function:

A task is a type of assignment that shall be created by a student on HATA.

4.2.12.4. Subordinates:

4.2.12.4.1. **Name**:

The title that shall be assigned to an assignment or task.

4.2.12.4.2. **Course**:

The class or course that shall be associated with a specific assignment or task.

4.2.12.4.3. **Notes**:

The notes entered by the user that shall pertain to a specific assignment or task.

4.2.12.4.4. **DueDate**:

The date (month and day) and time (hours, minutes, and seconds) that an assignment or task shall be completed.

4.2.12.4.5. **ReminderTime**:

An attribute of a task or assignment that is the amount of time remaining from the task due date to the present time, that a student shall be sent a notification that the task is due.

4.2.12.4.6. Status:

Whether or not an assignment or task is completed.

4.2.12.5. Dependencies:

4.2.12.5.1. None

4.2.12.6. Interfaces:

4.2.12.6.1. None

4.2.12.7. Data:

4.2.12.7.1. Task name; course; notes; due date; reminder time; status of a task

4.2.13. CloudStorageController

4.2.13.1. Component Identifier:

CloudStorage

4.2.13.2. Purpose:

This component shall be used to store data that shall be synced with HATA.

4.2.13.3. Function:

The location where all app settings and information shall be stored on the Cloud.

4.2.13.4. Subordinates:

4.2.13.4.1. AppData:

The information that the mobile app shall use to provide its intended functionality.

4.2.13.4.2. UserAccount:

The link that shall associate HATA app data with a specific mobile device.

4.2.13.5. Dependencies:

4.2.13.5.1. Network connectivity (via Wi-Fi or mobile network)

4.2.13.6. Interfaces:

4.2.13.6.1. ServiceNetworkCommunicationController

4.2.13.6.2. MobileNetworkCommunicationController

4.2.13.6.3. InternalStorageController

4.2.13.7. Data:

4.2.13.7.1. App settings; assignment and task data.

4.2.14. UserAccountModel

4.2.14.1. Component Identifier:

UserAccount

4.2.14.2. Purpose:

This component shall link the HATA app data with the mobile device.

4.2.14.3. Function:

The link that shall associate HATA app data with a specific mobile device.

4.2.14.4. Subordinates:

4.2.14.4.1. **Username**:

An identifying title unique to a specific user.

4.2.14.4.2. **Password:**

A user specified string that shall be associated with a username to grant users access to the app data.

4.2.14.5. Dependencies:

4.2.14.5.1. Username and password

4.2.14.6. Interfaces:

4.2.14.6.1. There shall be interfaces for account creation and resetting passwords that shall interact with this component.

4.2.14.7. Data:

4.2.14.7.1. Username; password

4.2.15. ServiceNetworkCommunicationController

4.2.15.1. Component Identifier:

ServiceNetworkCommunication

4.2.15.2. Purpose:

This component shall pass information to and from the web service securely.

4.2.15.3. Function:

How the HATA mobile application and the web service shall communicate.

4.2.15.4. Subordinates:

4.2.15.4.1. Appdata:

The information that the mobile app shall use to provide its intended functionality.

4.2.15.5. Dependencies:

4.2.15.5.1. This shall have constraints placed upon it so that it only attempts to synchronize with the HATA web service when there is a connection.

4.2.15.6. Interfaces:

4.2.15.6.1. MobileNetworkCommunicationController

4.2.15.6.2. InternalStorageController

4.2.15.7. Data:

4.2.15.7.1. Stored data for the application

4.2.16. MobileNetworkCommunicationController

4.2.16.1. Component Identifier:

MobileNetworkCommunication

4.2.16.2. Purpose:

This component shall establish a connection between the HATA mobile application and the web service.

4.2.16.3. Function:

The network communication that shall be used to synchronize with the web service.

4.2.16.4. Subordinates:

4.2.16.4.1. Wi-FiConnection:

This is a technology that shall allow a device to send and receive information through a wireless local area network (WLAN).

4.2.16.4.2. MobileDataConnection:

The communication mechanism where information shall be both sent and received through the cell phone communication transmitter on a mobile device.

4.2.16.4.3. MobileDataUsageSetting:

Whether or not HATA shall use a mobile data connection for synchronizing in the absence of Wi-Fi connectivity.

4.2.16.4.4. ConnectionErrorMessage:

A window shall appear on top of the current application window alerting the user of failure to synchronize with the web service due to lack of network connectivity stating, "Unable to sync. Network not available. Please try again later."

4.2.16.5. Dependencies:

4.2.16.5.1. This shall depend only upon the permission given to the HATA mobile application to use mobile data.

4.2.16.6. Interfaces:

4.2.16.6.1. InternalStorageController

4.2.16.7. Data:

4.2.16.7.1. Permissions given to the application to use mobile data

4.3. Account Management

4.3.1. Component Identifier:

AccountManagement

4.3.2. Purpose:

This component shall allow the user to manage User account information.

4.3.3. Function:

Manages User Account creation, view, information, and reuse.

4.3.4. Subordinates:

4.3.4.1. AccountInformationController

4.3.4.2. AccountInformationView

4.3.4.3. EasyAccessModePrompt

4.3.4.4. FirstTimeUser

4.3.4.5. LoginCreationController

4.3.4.6. LoginCreationView

4.3.4.7. LoginScreenController

4.3.4.8. LoginScreenView

4.3.4.9. ProfilePicture

4.3.4.10. SettingsPageController

4.3.4.11. SettingsPageView

4.3.5. Dependencies:

4.3.5.1. User input

4.3.5.2. Changes to the UserSettingsModel and UserAccountModel.

4.3.5.3. Dependant upon the mobile application to run.

4.3.6. Interfaces:

4.3.6.1. Works directly with UserSettingsModel, UserAccountModel.

4.3.7. Data:

4.3.7.1. User input; saved user settings

4.3.8. AccountInformationController

4.3.8.1. Component Identifier:

AccountInformationController

4.3.8.2. Purpose:

This component shall be used to view and update account user information.

4.3.8.3. Function:

This controller shall allow for the viewing and editing of account information.

4.3.8.4. Subordinates:

4.3.8.4.1. BackgroundDataGrabber

4.3.8.5. Dependencies:

4.3.8.5.1. User input

4.3.8.5.2. Any changes to user information or settings.

4.3.8.6. Interfaces:

4.3.8.6.1. AccountInformationView

4.3.8.6.2. EasyAccessModePrompt

4.3.8.6.3. LoginCreationController

4.3.8.6.4. LoginCreationView

4.3.8.6.5. LoginScreenController

4.3.8.6.6. LoginScreenView

4.3.8.6.7. SettingsPageController

4.3.8.6.8. SettingsPageView

4.3.8.7. Data:

4.3.8.7.1. Personal information; school information

4.3.9. AccountInformationView

4.3.9.1. Component Identifier:

AccountInformationView

4.3.9.2. Purpose:

This component shall have a view where all account information is visible.

4.3.9.3. Function:

The view shall show the user's account information and where the account is linked to Facebook and Google+.

4.3.9.4. Subordinates:

4.3.9.4.1. Profile Picture

4.3.9.4.2. Username

4.3.9.4.3. Contact Info

4.3.9.4.4. Notifications settings

4.3.9.4.5. Alarms

4.3.9.4.6. Linking to Social Media Accounts

4.3.9.4.7. Background Refresh Toggle Slider

4.3.9.5. Dependencies:

4.3.9.5.1. AccountInformationController

4.3.9.6. **Interfaces**:

4.3.9.6.1. AccountInformationController

4.3.9.6.2. EasyAccessModePrompt

4.3.9.6.3. LoginCreationController

4.3.9.6.4. LoginCreationView

4.3.9.6.5. LoginScreenController

4.3.9.6.6. LoginScreenView

4.3.9.6.7. SettingsPageController

4.3.9.6.8. SettingsPageView

4.3.9.7. **Data**:

4.3.9.7.1. AccountInformation; LoginCreation; SettingsPage

4.3.10. EasyAccessModePrompt

4.3.10.1. **Component Identifier**:

EasyAccessModePrompt

4.3.10.2. **Purpose**:

This component shall notify the user upon first login that there is an option to login without inputting login credentials after the initial user login.

4.3.10.3. **Function**:

The function shall allow the user the ability to bypass the login screen and get into the HATA without inputting login credentials.

4.3.10.4. **Subordinates**:

4.3.10.4.1. Single Sign On

4.3.10.4.2. EasyAccessModeOn

4.3.10.4.3. EasyAccessModeOff

4.3.10.4.4. FirstTimeLoginValidator

4.3.10.5. **Dependencies**:

4.3.10.5.1. User selection

4.3.10.5.2. Option input

4.3.10.6. Interfaces:

4.3.10.6.1. LoginScreenView

4.3.10.7. Data:

4.3.10.7.1. AccountInformation; LoginCreation; SettingsPage.

4.3.11. FirstTimeUser

4.3.11.1. Component Identifier:

First Time User

4.3.11.2. Purpose:

This component shall assist the user in logging in for the first time.

4.3.11.3. Function:

This person shall log in for the first time.

4.3.11.4. Subordinates:

4.3.11.4.1. Username

4.3.11.4.2. Password

4.3.11.5. Dependencies:

4.3.11.5.1. The user shall create I-Learn login credentials prior to logging on.

4.3.11.6. Interfaces:

4.3.11.6.1. AccountInformationView

4.3.11.6.2. EasyAccessModePrompt

4.3.11.6.3. LoginCreationView

4.3.11.6.4. LoginScreenView

4.3.11.6.5. Profile Picture

4.3.11.6.6. SettingsPageView

4.3.11.7. Data:

4.3.11.7.1. Username; password; personal information; school information

4.3.12. LoginCreationController

4.3.12.1. Component Identifier:

LoginCreationController

4.3.12.2. Purpose:

This component shall verify that valid and acceptable usernames and passwords are created.

4.3.12.3. Function:

This controller shall create the login information. The LoginCreationController shall validate both the username and the password to make sure that they are valid and fit all criteria.

This controller shall send out a confirmation email of the login information. If the user inputs an incorrect or invalid username the LoginCreationController shall prompt the user to enter a valid one.

4.3.12.4. Subordinates:

4.3.12.4.1. UserNameValidator

4.3.12.4.2. PasswordValidator

4.3.12.4.3. Confirmation Email

4.3.12.4.4. ValidUserNamePromptMessage

4.3.12.5. Dependencies:

4.3.12.5.1. User Input

4.3.12.6. Interfaces:

4.3.12.6.1. LoginCreationView

4.3.12.7. Data:

4.3.12.7.1. Personal Data; login credential selections

4.3.13. LoginCreationView

4.3.13.1. Component Identifier:

LoginCreationView

4.3.13.2. Purpose:

This component shall allow the user to view and select login credentials.

4.3.13.3. Function:

This view shall explain to the user where and what to enter to set up a login.

4.3.13.4. Subordinates:

4.3.13.4.1. Profile Picture

4.3.13.4.2. Input Fields

4.3.13.5. Dependencies:

4.3.13.5.1. User Input

4.3.13.5.2. LoginCreationController

4.3.13.6. Interfaces:

4.3.13.6.1. LoginScreenView

4.3.13.7. Data:

4.3.13.7.1. Username; password

4.3.14. LoginScreenController

4.3.14.1. Component Identifier:

LoginScreenController

4.3.14.2. Purpose:

This component shall validate login credentials.

4.3.14.3. Function:

The LoginScreenController function shall be to validate the username and password and provide an invalid username error message if an incorrect username and password are entered.

To allow for the use of the OAuth API's from sites such as Google+ and Facebook.

4.3.14.4. Subordinates:

4.3.14.4.1. OAuthAPI

4.3.14.4.2. UsernameValidator

4.3.14.4.3. PasswordValidator

4.3.14.4.4. ValidUsernamePromptMessage

4.3.14.5. Dependencies:

4.3.14.5.1. Input from User

4.3.14.5.2. LoginScreenView

4.3.14.6. Interfaces:

4.3.14.6.1. LoginScreenView

4.3.14.7. Data:

4.3.14.7.1. Username; password

4.3.15. LoginScreenView

4.3.15.1. Component Identifier:

LoginScreenView

4.3.15.2. Purpose:

This component shall provide a place for the user to input login credentials and provide information to the user if the inputted login credentials are considered invalid.

4.3.15.3. Function:

The view shall provide a location for the login credentials to be input. The view shall also display a message that the student may login using Facebook, Google+, and I-Learn.

4.3.15.4. Subordinates:

4.3.15.4.1. NotificationMessage

4.3.15.4.2. Profile Picture

4.3.15.4.3. UsernameInputField

4.3.15.4.4. PasswordInputField

4.3.15.5. Dependencies:

4.3.15.5.1. LoginScreenController

4.3.15.6. Interfaces:

4.3.15.6.1. LoginScreenView

4.3.15.6.2. EasyAccessMode Prompt

4.3.15.7. Data:

4.3.15.7.1. Username; password

4.3.16. ProfilePicture

4.3.16.1. Component Identifier:

ProfilePicture

4.3.16.2. Purpose:

The ProfilePicture shall allow the user to customize their profile.

4.3.16.3. Function:

The profile picture shall allow the user to customize and personalize their profile.

4.3.16.4. Subordinates:

4.3.16.4.1. DefaultProfilePicture

4.3.16.4.2. StoredProfilePicture

4.3.16.5. Dependencies:

4.3.16.5.1. Picture uploaded on SettingsPage.

4.3.16.6. Interfaces:

4.3.16.6.1. AccountInformationView

4.3.16.6.2. EasyAccessModePrompt

4.3.16.6.3. LoginCreationView

4.3.16.6.4. LoginScreenView

4.3.16.6.5. ProfilePicture

4.3.16.6.6. SettingsPageView

4.3.16.7. Data:

4.3.16.7.1. .png; .jpg; .jpeg

4.3.17. SettingsPageController

4.3.17.1. Component Identifier:

SettingsPageController

4.3.17.2. Purpose:

This component shall allow the user to view and modify settings.

4.3.17.3. Function:

Shall allow the user the change settings in HATA.

Customizable settings shall include notifications and sounds.

Shall allow picture file uploads.

Profile pictures shall be resized automatically, based on available screen size, by this controller.

4.3.17.4. Subordinates:

- **4.3.17.4.1.** File upload
- **4.3.17.4.2.** ProfilePictureEditor
- **4.3.17.4.3.** FileTypeValidator
- **4.3.17.4.4.** ImageSizeScaler
- **4.3.17.4.5.** NotificationConfiguration
- **4.3.17.4.6.** SoundConfiguration
- **4.3.17.4.7.** BackgroundDataGrabber

4.3.17.5. Dependencies:

- **4.3.17.5.1.** SettingsPageView

4.3.17.6. Interfaces:

- **4.3.17.6.1.** SettingsPageView
- **4.3.17.6.2.** AccountInformationView
- **4.3.17.6.3.** AccountInformationController
- **4.3.17.6.4.** EasyAccessModePrompt
- **4.3.17.6.5.** LoginCreationController
- **4.3.17.6.6.** LoginCreationView
- **4.3.17.6.7.** LoginScreenController
- **4.3.17.6.8.** LoginScreenView
- **4.3.17.6.9.** SettingsPageController
- **4.3.17.6.10.** SettingsPageView

4.3.17.7. Data:

- **4.3.17.7.1.** User Settings from SettingsPageController; ProfilePicture

4.3.18. SettingsPageView

4.3.18.1. Component Identifier:

SettingsPageView

4.3.18.2. Purpose:

This component shall allow users to view and modify settings.

4.3.18.3. Function:

This view shall allow the user to be able to view and change settings in regards to HATA.

4.3.18.4. Subordinates:

4.3.18.4.1. ProfilePicture

4.3.18.4.2. Notifications

4.3.18.4.3. Sounds

4.3.18.4.4. Account Information

4.3.18.5. Dependencies:

4.3.18.5.1. SettingsPageController

4.3.18.6. Interfaces:

4.3.18.6.1. SettingsPageController

4.3.18.7. Data:

4.3.18.7.1. User Settings from SettingsPageController

4.4. AppGUI

4.4.1. Component Identifier:

AppGUI

4.4.2. Purpose:

This component shall act as HATA's graphical user interface, which the student shall utilize to access and interact with all functions of the system.

4.4.3. Function:

System package that holds all of the presentation layer for the application.

4.4.4. Subordinates:

4.4.4.1. AutomaticColorCoding

4.4.4.2. SearchDisplayResults

4.4.4.3. HomePageEntity

4.4.4.4. TaskCreationScreen

4.4.5. Dependencies:

4.4.5.1. Dependent upon the entirety of the system.

4.4.6. Interfaces:

4.4.6.1. ClientSynchronization

4.4.6.2. AccountManagement

4.4.6.3. Options

4.4.7. Data:

All HATA data will be used and presented by the AppGUI

4.4.8. AutomaticColorCoding

4.4.8.1. Component Identifier:

AutomaticColorCoding

4.4.8.2. Purpose:

AutomaticColorCoding allows for a way to sort and see the prioritization of assignments and tasks.

4.4.8.3. Function:

Colors may be customized or set as automatic defaults for selected categories.

4.4.8.4. Subordinates:

4.4.8.4.1. ColorList:

The model shall contain a list of colors where each color can be redefined with a custom color.

4.4.8.4.2. PriorityLevels:

Assignments that don't match the selected filter color shall be hidden. The colors shall be: red for high, yellow for medium, and green for low.

4.4.8.4.3. DueDates:

All assignments or filtered assignments (if filter is selected), should be sorted by due date in ascending order.

4.4.8.5. Dependencies:

4.4.8.5.1. The Assignment List shall be required to be populated with at least one task or assignment in order to be sorted by color.

4.4.8.6. Interfaces:

4.4.8.6.1.
Shall use AssignmentList to give each assignment label a color (see diagram 28) that can be used to sort assignments by course, priority, week, and type. The sort state shall not persist beyond each viewing.

4.4.8.7. Data:

4.4.8.7.1.
Customized colors shall be locally stored on the device

4.4.9. SearchDisplayResults

4.4.9.1. Component Identifier:

SearchDisplayResults

4.4.9.2. Purpose:

This component shall allow for entry of search criteria and present the search results to the student.

4.4.9.3. Function:

This view shall filter tags and text-based tags by color priority and ascending due date.

4.4.9.4. Subordinates:

4.4.9.4.1. AssignmentsTasksList:

Every assignment and task that matches the search criteria shall appear in a list sorted by due date.

4.4.9.4.2. AssignmentsTasksLink:

Every assignment and task that matches the search criteria shall be linked, providing direct access to the associated assignment or task.

4.4.9.4.3. SearchBar:

A search bar with a search icon shall allow for entry of search criteria.

4.4.9.5. Dependencies:

4.4.9.5.1.
Student Information

4.4.9.5.2.
Class information

4.4.9.5.3.
Task Information

4.4.9.5.4. Assignment information

4.4.9.6. Interfaces:

4.4.9.6.1. None.

4.4.9.7. Data:

4.4.9.7.1. An existing assignment or task

4.4.10. HomePageEntity

4.4.10.1. Component Identifier:

HomePageEntity

4.4.10.2. Purpose:

This component shall display the home page of HATA.

4.4.10.3. Function:

The HomePageEntity gets the home page view and displays it on the mobile device.

4.4.10.4. Subordinates:

4.4.10.4.1. HomePageController:

The HomePageController shall handle the creation and updating of the home page.

4.4.10.4.2. HomePageView:

The HomePageView shall handle the display of the home page to the student.

4.4.10.4.3. ProgressBar:

The ProgressBar shall display a graphical bar showing the progress of I-Learn synchronization.

4.4.10.4.4. AssignmentList:

The assignment list shall display a list of assignments in the home page.

4.4.10.5. Dependencies:

4.4.10.5.1. None

4.4.10.6. Interfaces:

4.4.10.6.1. None

4.4.10.7. Data:

4.4.10.7.1. None

4.4.11. AssignmentList

4.4.11.1. Component Identifier:

AssignmentList

4.4.11.2. Purpose:

This component shall represent a list of tasks and assignments.

4.4.11.3. Function:

Every assignment label shall have a color that can be used to sort assignments by course, priority, week, and type.

All assignments will be listed up to two weeks into the future.

4.4.11.4. Subordinates:

4.4.11.4.1. Assignments

4.4.11.5. Dependencies:

4.4.11.5.1. The Assignment List shall be required to be populated with at least one task or assignment in order to be sorted by color.

4.4.11.6. Interfaces:

4.4.11.6.1. AutomaticColorCoding

4.4.11.7. Data:

4.4.11.7.1. Course data including date and time for Assignment due dates

4.4.12. Assignment

4.4.12.1. Component Identifier:

Assignment

4.4.12.2. Purpose:

This component represent an objective the student shall complete**.**

4.4.12.3. Function:

Contain information for student objectives, including due date, time needed to complete, and assignment description.

4.4.12.4. Subordinates:

4.4.12.4.1. Due date

4.4.12.4.2. Estimated time to completion.

4.4.12.4.3. Assignment description

4.4.12.5. Dependencies:

4.4.12.5.1. None

4.4.12.6. Interfaces:

4.4.12.6.1. Tasks

4.4.12.7. Data:

4.4.12.7.1. Due date and time; estimated time to completion; description

4.4.13. Task

4.4.13.1. Component Identifier:

Task

4.4.13.2. Purpose:

This component represents an objective created by the student that shall be completed. It is a subclass of Assignment.

4.4.13.3. Function:

Represents an objective needed to be completed.

4.4.13.4. Subordinates:

4.4.13.4.1. Due date

4.4.13.4.2. Time to complete

4.4.13.4.3. Description

4.4.13.5. Dependencies:

4.4.13.5.1. None

4.4.13.6. Interfaces:

4.4.13.6.1. Assignment

4.4.13.7. Data:

4.4.13.7.1. Due date and time; estimated time to completion; description

4.4.14. NavigationTabs

4.4.14.1. Component Identifier:

NavigationTabs

4.4.14.2. Purpose:

This component shall contain all of the container views for the Home Page and enables switching between the views as the student selects the tabs. These tabs will be located at the top of the HomePage.

4.4.14.3. Function:

Provide tabular links to different container views for the Home Page.

4.4.14.4. Subordinates:

4.4.14.4.1. Tab

4.4.14.5. Dependencies:

4.4.14.5.1. HomePageView

4.4.14.5.2. HomePageController

4.4.14.6. Interfaces:

4.4.14.6.1. ProgressBar

4.4.14.7. Data:

4.4.14.7.1. None

4.4.15. CourseView

4.4.15.1. Component Identifier:

CourseView

4.4.15.2. Purpose:

This component shall display the course and its associated information within HATA.

4.4.15.3. Function:

Takes information from the course and displays it to the user in the AppGUI.

4.4.15.4. Subordinates:

4.4.15.4.1. None

4.4.15.5. Dependencies:

4.4.15.5.1. None

4.4.15.6. Interfaces:

4.4.15.6.1. None

4.4.15.7. Data:

4.4.15.7.1. Course Information

4.4.16. ProgressBar

4.4.16.1. Component Identifier:

ProgressBar

4.4.16.2. Purpose:

This component shall display the current progress of background tasks.

4.4.16.3. Function:

Create a representation of the user's progress in the form of a bar filling up as the user completes required tasks.

4.4.16.4. Subordinates:

4.4.16.4.1. None

4.4.16.5. Dependencies:

4.4.16.5.1. AssignmentList

4.4.16.5.2. Tasks

4.4.16.6. Interfaces:

4.4.16.6.1. Shall use ClientSynchronization to get sync progress with HATAWebAPI.

4.4.16.7. Data:

4.4.16.7.1. Assignment information; Task information

4.4.17. CalendarView

4.4.17.1. Component Identifier:

CalendarViewController

4.4.17.2. Purpose:

This component shall display information for the entire month listed in each date block by due date in ascending order of reminder time.

4.4.17.3. Function:

The CalendarView shall allow direct access to the Monthly, Weekly, and Daily calendar views to see more information about an assignment or task, such as the course, due date, and specific time of day the assignment or task is due. All assignments and tasks shall be listed in each date block with the associated assignment or task name and due date.

4.4.17.4. Subordinates:

4.4.17.4.1. CalendarNavigationTabs:

HATA shall display Monthly, Weekly, and Daily Navigation tabs, located at the top of the calendar views. Selecting the tab with the associated view name shall direct the user to that view.

4.4.17.4.2. WeeklyCalendarView:

The weekly view shall display information for the entire week when the user selects the weekly tab from the top of the calendar view.

4.4.17.4.3. DailyCalendarView:

The daily view shall display information for the entire day when the student selects the daily tab from the top of the calendar view.

4.4.17.4.4. DefaultMonthlyView:

The monthly view shall display information for the entire month when selected by student, and is default display if no other calendar view else is selected.

4.4.17.5. Dependencies:

4.4.17.5.1. Calendar items shall be able to be created regardless of network connectivity. Any calendar items created while not connected to the internet will be persisted to the data access layer upon connection.

4.4.17.6. Interfaces:

4.4.17.6.1. None

4.4.17.7. Data:

4.4.17.7.1. Calendar items shall be stored as entries in the database; task attributes will be represented by individual columns in the database.

4.4.18. NotificationsView

4.4.18.1. Component Identifier:

NotificationsView

4.4.18.2. Purpose:

This component shall display a notification with task or assignment details to the user.

4.4.18.3. Function:

Shows notification set for a task or assignment.

4.4.18.4. Subordinates:

4.4.18.4.1. Notification

4.4.18.4.2. Due Date

4.4.18.4.3. Due Time

4.4.18.5. Dependencies:

4.4.18.5.1. Assignment

4.4.18.5.2. Task

4.4.18.6. Interfaces:

4.4.18.6.1. None

4.4.18.7. Data:

4.4.18.7.1. Assignment Information

4.4.18.7.2. Task Information

4.4.18.7.3. Saved Notification data including due date and time.

4.4.19. Notification

4.4.19.1. Component Identifier:

Notification

4.4.19.2. Purpose:

A class that represents a notification.

4.4.19.3. Function:

Contains and updates information including due date and time related to an assignment or task.

4.4.19.4. Subordinates:

4.4.19.4.1. DueDate

4.4.19.4.2. DueTime

4.4.19.5. Dependencies:

4.4.19.5.1. Assignment

4.4.19.5.2. Task

4.4.19.6. Interfaces:

4.4.19.6.1. None

4.4.19.7. Data:

4.4.19.7.1. DueDate; DueTime

4.4.20. AssignmentsViewEntity

4.4.20.1. Component Identifier:

AssignmentsViewController

4.4.20.2. Purpose:

This component shall represent an HTML view of the Assignment List.

4.4.20.3. Function:

This view shall display all of the information for a specific assignment or task when the student selects an assignment or task.

4.4.20.4. Subordinates:

4.4.20.4.1. **AssignmentsListModel:**

The AssignmenstListModel should import student information to populate the assignments associated with the student and displays the assignments on the assignment list.

4.4.20.4.2. **AssignmentsListView:**

The AssignmentsListView shall display a list of tasks or assignments by due date in ascending order and by course.

4.4.20.4.3. AssignmentsListController:

The AssignmentsListController shall receives from the model the list of tasks or assignments.

4.4.20.5. Dependencies:

4.4.20.5.1. None

4.4.20.6. Interfaces:

4.4.20.6.1. TaskCreationScreenEntity

4.4.20.7. Data:

4.4.20.7.1. None

4.4.21. TaskCreationScreenEntity

4.4.21.1. Component Identifier:

TaskCreationScreen

4.4.21.2. Purpose:

This component shall represent an HTML view of the Task Creation screen within HATA.

4.4.21.3. Function:

This view shall receive data from the student and passes this to the task repository.

4.4.21.4. Subordinates:

4.4.21.4.1. TaskCreationController:

Shall use the AssignmentsListController to receive from the model the list of tasks or assignments.

Shall add and persist tasks using the AssignmentListController.

Shall trigger the display of an icon when saving the task to indicate the task has been saved.

Shall display a notification on the TaskCreationView to the student if the task fails to save.

Shall trigger a notification on the TaskCreationView that editing is still enabled if the device loses network connectivity.

Shall redirect the student to the AssignmentsViewEntity when a task has been saved.

Shall block assignment information from being modified when a note is added.

Shall handle requests from TaskCreationView to revert assignment data to the original I-Learn contents.

4.4.21.4.2. TaskCreationView:

Shall display the options to create a task and accepts task information to pass to the TaskCreationController.

Shall display notification messages to the user that TaskCreationController triggers.

Shall display the task name, task course, task due date, task due time, task notes, task link, and the task reminder setting.

Shall use the mobile device's default date selector to set the due date.

Shall display a button to revert assignment data to the original I-Learn contents.

4.4.21.4.3. TaskInput:

Shall display voice and text input options to control input to the task.

4.4.21.5. Dependencies:

4.4.21.5.1. None

4.4.21.6. Interfaces:

4.4.21.6.1. Shall use ClientSynchronization to check for network availability to notify the user that their task data will be saved at a later point when network connectivity is resumed.

4.4.21.7. Data:

4.4.21.7.1. None

4.4.22. DailyOverviewNotifications

4.4.22.1. Component Identifier:

DailyOverviewSetting

4.4.22.2. Purpose:

This component shall send a display notification to alert the user of upcoming due dates on the current day.

4.4.22.3. Function:

This process shall get the assignments due for the current day and sends the due assignment information in a notification.

4.4.22.4. Subordinates:

4.4.22.4.1. DailyOverviewNotificationView:

This view shall display a summary of assignments due on the current day.

4.4.22.5. Dependencies:

4.4.22.5.1. None

4.4.22.6. Interfaces:

4.4.22.6.1. Notification

4.4.22.7. Data:

4.4.22.7.1. None

4.5. Options

4.5.1. Component Identifier:

Options

4.5.2. Purpose:

This component shall allow the student to set user-specific settings within HATA.

4.5.3. Function:

System package that holds all of the options and settings for the mobile application.

4.5.4. Subordinates:

4.5.4.1. NotificationSettings

4.5.4.2. DueDateSettings

4.5.4.3. TaskDeletingSettings

4.5.4.4. GroupWork

4.5.4.5. CompletedTasks

4.5.4.6. QuickTagFiltering

4.5.4.7. Assignment&TaskHiding

4.5.5. **Dependencies**:

4.5.5.1. Initial app setup

4.5.5.2. User input

4.5.6. **Interfaces**:

4.5.6.1. None

4.5.7. **Data**:

4.5.7.1. NotificationSettings; DueDateSettings; TaskDeletingSettings; GroupWork; CompletedTasks; QuickTagFiltering; Assignment&TaskHiding

4.5.8. **Notification Settings**

4.5.8.1. **Component Identifier:**

NotificationSettings

4.5.8.2. **Purpose:**

This component shall set when notifications should be sent to the student and what content shall be included.

4.5.8.3. **Function:**

These settings shall handle the timing of when notifications are sent and what is sent.

4.5.8.4. **Subordinates:**

4.5.8.4.1. **Assignments:**

An Assignment represents an objective the student shall complete**.**

4.5.8.4.2. **Tasks:**

A Task represents an objective created by the student that shall be completed.

4.5.8.4.3. **NotificationDate:**

A Notification Date represents the date (month and day) and time (hours, minutes, and seconds) that a notification will be created.

4.5.8.5. Dependencies:

4.5.8.5.1. TaskReminderSetting

4.5.8.5.2. SMSNotificationSetting

4.5.8.5.3. Notification

4.5.8.6. Interfaces:

4.5.8.6.1. InternalStorageController

4.5.8.6.2. CloudStorageController

4.5.8.7. Data:

4.5.8.7.1. Assignment or Task; NotificationDate

4.5.9. Due Date Settings

4.5.9.1. Component Identifier:

DueDateSettings

4.5.9.2. Purpose:

This component shall display settings for assignment and task due dates.

4.5.9.3. Function:

The DueDateSettings shall set the amount of time required for an assignment or task to appear in the upcoming list of assignments.

4.5.9.4. Subordinates:

4.5.9.4.1. **Assignments**:

An Assignment represents an objective the student shall complete**.**

4.5.9.4.2. **Tasks:**

A Task represents an objective created by the student that shall be completed.

4.5.9.4.3. **Due Date:**

A Due Date represents the date (month and day) and time (hours, minutes, and seconds) that an assignment or task shall be completed.

4.5.9.5. Dependencies:

4.5.9.5.1. None

4.5.9.6. Interfaces:

4.5.9.6.1. InternalStorageController

4.5.9.6.1.1. CloudStorageController

4.5.9.7. Data:

4.5.9.7.1. Assignments or Tasks; DueDate

4.5.10. Task Deleting Settings

4.5.10.1. Component Identifier:

TaskDeletingSettings

4.5.10.2. Purpose:

This component shall display settings for task deletions.

4.5.10.3. Function:

The TaskDeletingSettings shall set the user preference for task deletion time as well as to archive or completely remove a deleted task.

4.5.10.4. Subordinates:

4.5.10.4.1. **Assignments**:

An Assignment represents an objective the student shall complete**.**

4.5.10.4.2. **Tasks:**

A Task represents an objective created by the student that shall be completed.

4.5.10.5. Dependencies:

4.5.10.5.1. None

4.5.10.6. Interfaces:

4.5.10.6.1. InternalStorageController

4.5.10.6.2. CloudStorageController

4.5.10.7. Data:

4.5.10.7.1. Assignments or Tasks

4.5.11. Group Work

4.5.11.1. Component Identifier:

GroupWorkSettings

4.5.11.2. Purpose:

This component shall display settings for sharing assignment or task information amongst group members.

4.5.11.3. Function:

The GroupWork option shall display settings regarding whether assignments and tasks can be shared amongst group members.

4.5.11.4. Subordinates:

4.5.11.4.1. **Assignments**:

An Assignment represents an objective the student shall complete**.**

4.5.11.4.2. **Tasks:**

A Task represents an objective created by the student that shall be completed.

4.5.11.4.3. **GroupAssignment**

4.5.11.4.4. **GroupTask**

4.5.11.5. Dependencies:

4.5.11.5.1. None

4.5.11.6. Interfaces:

4.5.11.6.1. InternalStorageController

4.5.11.6.2. CloudStorageController

4.5.11.7. Data:

4.5.11.7.1. GroupAssignment; GroupTalk; either Assignments or Tasks.

4.5.12. Completed Tasks

4.5.12.1. Component Identifier:

CompletedTaskSettings

4.5.12.2. Purpose:

This component shall display settings for completed assignments and tasks.

4.5.12.3. Function:

The CompletedTaskSettings shall display settings for removal of completed assignments when an assignment has been completed and the deletion date has passed.

4.5.12.4. Subordinates:

4.5.12.4.1. Tasks:

A Task represents an objective created by the student that shall be completed.

4.5.12.4.2. Assignments:

An Assignment represents an objective the student shall complete.

4.5.12.5. Dependencies:

4.5.12.5.1. None

4.5.12.6. Interfaces:

4.5.12.6.1. InternalStorageController

4.5.12.6.2. CloudStorageController

4.5.12.7. Data:

4.5.12.7.1. Assignments; Tasks

4.5.13. Quick Tag Filtering

4.5.13.1. Component Identifier:

QuickTagFilteringSettings

4.5.13.2. Purpose:

This component shall implement a tagging system and allow the user to create their own tags for assignments and tasks.

4.5.13.3. Function:

The QuickTagFiltering settings shall allow the student to choose whether or not to enable a tagging system for assignments and tasks.

4.5.13.4. Subordinates:

4.5.13.4.1. QuickTagView:

The QuickTagView shall display a list of tags.

4.5.13.4.2. QuickTagController:

The QuickTagController shall receive from the model the list of tags.

4.5.13.4.3. QuickTagCreationView:

The QuickTagCreationView shall display a screen for creation of a quick tag.

4.5.13.4.4. QuickTagCreationController:

The QuickTagCreationController shall receive from the model the data for creating a new tag.

4.5.13.4.5. DefaultTagController:

The DefaultTagController shall set the default tag for an assignment or task.

4.5.13.5. Dependencies:

4.5.13.5.1. None

4.5.13.6. Interfaces:

4.5.13.6.1. InternalStorageController

4.5.13.6.2. CloudStorageController

4.5.13.7. Data:

4.5.13.7.1. QuickTagController; QuickTagCreationController,; DefaultTagController.

4.5.14. Assignment & Task Hiding

4.5.14.1. Component Identifier:

AssignmentTaskHidingSettings

4.5.14.2. Purpose:

This component shall determine if an assignment or task is to be hidden.

4.5.14.3. Function:

This option setting shall allow the student to hide an assignment or task from displaying in HATA.

4.5.14.4. Subordinates:

4.5.14.4.1. HidingConfirmationView:

The HidingConfirmationView shall display a prompt screen before hiding an assignment or task.

4.5.14.4.2. HidingConfirmationController:

The HidingConfirmationController shall receive from the model the appropriate assignment or task data corresponding to the assignment

4.5.14.4.3. DisableFutureConfirmationsView:

The DisableFutureConfirmationView shall display an option to hide future prompt screens before hiding an assignment or task.

4.5.14.4.4. DisableFutureConfirmationsController:

The DisableFutureConfirmationsController shall receive from the model the global option of whether prompts are to be hidden or not.

4.5.14.4.5. HidingSynchronizationController:

The HidingSynchronizationController shall receive data from the Assignments or Tasks model to determine if a synchronization will exclude or include the assignment or task.

4.5.14.5. Dependencies:

4.5.14.5.1. None

4.5.14.6. Interfaces:

4.5.14.6.1. InternalStorageController

4.5.14.6.2. CloudStorageController

4.5.14.7. Data:

4.5.14.7.1. HidingConfirmationController;
DisableFutureConfirmationsController;
HidingSynchronizationController

4.6. HATAWebAPI

4.6.1. Component Identifier:

HATAWebAPI

4.6.2. Purpose:

This component shall act as HATA's back end service to provide assignment synchronization for the user.

4.6.3. Function:

This component shall cover the web server and backend web service.

4.6.4. Subordinates:

4.6.4.1. Tasks

4.6.4.2. Assignment Lists

4.6.4.3. Security and Authentication

4.6.4.4. Ilearn Synchronizer

4.6.4.5. Settings Controller

4.6.4.6. Data Layer

4.6.5. Dependencies:

4.6.5.1. None

4.6.6. Interfaces:

4.6.6.1. (See Appendix 7.15 Diagram 15)

4.6.7. Data:

4.6.7.1. None

4.6.8. Tasks

4.6.8.1. Component Identifier:

TaskController

4.6.8.2. Purpose:

This component shall provide endpoints for interacting with task objects. Tasks shall be able to be created, modified, and deleted.

4.6.8.3. **Function**:

This entity shall receive data from the user and pass this to the task repository. The task repository shall then decide where to persist this data to, depending on network connectivity.

4.6.8.4. **Subordinates:**

4.6.8.4.1. **CreateTask:**

When the user creates a new task this subordinate shall activate. This function shall create a new task instance and pass it to the task repository.

4.6.8.4.2. **FetchTask:**

FetchTask shall pass the id of a desired task to the task repository, expecting the task object to be returned. This task object shall then be returned by FetchTask

4.6.8.4.3. **DeleteTask:**

DeleteTask shall send the id of a selected task to the task repository, which will handle the deletion of said task.

4.6.8.4.4. **EditTask:**

EditTask shall call FetchTask to get a desired task object. This object shall then be updated to reflect the data input by the user and passed to the task repository to be persisted.

4.6.8.4.5. **TaskRepository:**

This shall handle the persisting/fetching of data to/from the data access layer. The TaskRepository will decide to persist/fetch data to/from local storage or through syncing with I-Learn depending on internet connectivity.

4.6.8.5. **Dependencies**:

4.6.8.5.1. Tasks shall be able to be created regardless of network connectivity. Any tasks created while not connected to the internet shall be persisted to the data access layer upon connection.

4.6.8.6. I**nterfaces:**

4.6.8.6.1. Data Layer

4.6.8.7. **Data**:

4.6.8.7.1. Task objects shall be stored as entries in the database; task attributes shall be represented by individual columns in the database

4.6.9. Assignment Lists

4.6.9.1. Component Identifier:

AssignmentsController

4.6.9.2. Purpose:

This component shall fetch assignments from the I-Learn using the Valence API. These assignments shall return to the view for the user.

4.6.9.3. Function:

This entity shall call out to Brightspace using the Valence API to request assignments to be stored in the database

4.6.9.4. Subordinates:

4.6.9.4.1. FetchAllAssignments:

This shall request all the user's assignments from the Valence API and returns them to the view. It shall also pass the fetched assignments to the AssignmentRepository.

4.6.9.4.2. **FetchAssignmentsBasedOnDateRange**:

This shall request all the user's assignments using the Valence API that are due within a specified date range and returns these to the view. It shall also pass the fetched assignments to the AssignmentRepository.

4.6.9.4.3. FetchAssignmentsBasedOnCourse:

This shall request all the user's assignments using the Brightspace API that belong to a certain course and returns them to the view. It shall also pass the fetched assignments to the AssignmentRepository.

4.6.9.4.4. AssignmentRepository:

This shall persist assignments to data layer for offline fetching.

4.6.9.5. Dependencies:

4.6.9.5.1.
Internet connectivity shall be required for this component to function. Valid I-Learn credentials shall be required for this component to function.

4.6.9.6. Interfaces:

4.6.9.6.1.
Data Layer

4.6.9.7. Data:

4.6.9.7.1.
Assignments shall be stored as Assignment Objects and persisted to the database as entries in the assignments table

4.6.10. Security and Authentication

4.6.10.1. Component Identifier:

AuthController

4.6.10.2. Purpose:

This component shall handle the security and user authentication of HATA.

4.6.10.3. Function:

The AuthController of HATA shall register the student's email/username and password if they not already located in the HATA web services. This information shall be stored and encrypted. Once the student's registry is confirmed, then HATA shall use the student's credential information to setup the synchronization with I-Learn 3.0 databases and servers. HATA shall also assign a session token to the shall device.

4.6.10.4. Subordinates:

4.6.10.4.1. UserAction:

The UserAction shall contain functionality regarding the creation of a new student, the modification of an existing student, or the deletion of an existing student through the UserRepository.

4.6.10.4.2. UserRepository:

See 4.7.9 Data Layer.

4.6.10.4.3. UserAuthenticator:

This subordinate shall compare incoming passwords with the hashed password stored in the data layer. It shall use the UserRepository to access the data layer.

4.6.10.4.4. Security:

HATA shall use the Node.js crypto module for encryption and decryption of the student's password into and from the database.

4.6.10.4.5. Authentication:

The student shall authenticate by completing an HTML form. This will generate a session token that may be used for the ILearn Synchronizer. The session token shall be generated using the Node.js crypto module as well.

4.6.10.5. Dependencies:

4.6.10.5.1. HATA shall depend the student's information data that is found in HATA web services.

4.6.10.5.2. HATA shall depend on Node.js crypto module.

4.6.10.6. Interfaces:

4.6.10.6.1. User Repository

4.6.10.7. Data:

4.6.10.7.1. Username; encrypted password.

4.6.11. Ilearn Synchronizer

4.6.11.1. Component Identifier:

ILearnSync

4.6.11.2. Purpose:

This component shall handle the synchronization from the I-Learn 3.0 database to the HATA web service.

4.6.11.3. Function:

Once a student has been authenticated with HATA, the IlearnSync shall send requests to the I-Learn server so that I-Learn server may respond with a list of tasks and assignments to the HATA web service.

The list of tasks and assignments shall override any existing assignments.

4.6.11.4. Subordinates:

4.6.11.4.1. Authentication:

The HATA web service shall use a student's session token to start synchronizing with ILearn Servers. .

4.6.11.4.2. Synchronizer:

The synchronization shall be a list of GET methods via Expressjs web server architecture. This list of GET methods should be all the tasks and assignments that already exist in I-Learn. The GET methods shall utilize the Valence API that is maintained by Brightspace, and HATA web service shall send header data specifying MIME type and package kilobyte size.

4.6.11.5. Dependencies:

4.6.11.5.1. HATA shall depend on if the I-Learn servers respond with a timeout error.

4.6.11.5.2. In order to sync data from I-Learn 3.0, HATA shall depend on the student's authentication information.

4.6.11.5.3. HATA shall use REST web service calls to synchronize the data.

4.6.11.6. Interfaces:

4.6.11.6.1. UserRespository

4.6.11.7. Data:

4.6.11.7.1. Student's assignment list that has been synced from the I-Learn 3.0 database

4.6.12. Data Layer

4.6.12.1. Component Identifier:

DataAccessUtility

4.6.12.2. Purpose:

This component shall access the MySQL data layer.

4.6.12.3. Function:

The different entities of the HATAWebAPI shall call out to this utility, which shall retrieve data from the MySQL database. Sequelize, an ORM, shall be used to make calls out to the database.

4.6.12.4. Subordinates:

4.6.12.4.1. SettingsRepository:

This subordinate shall fetch and save data regarding settings into the database.

4.6.12.4.2. UserRepository:

This subordinate shall fetch and save data regarding users into the database.

4.6.12.4.3. TaskRepository:

This subordinate shall fetch and save data regarding tasks into the database.

4.6.12.5. Dependencies:

4.6.12.5.1. **DataAccessUtility** shall be dependent upon the MySQL server's resource limits for reading and writing data.

4.6.12.5.2. **DataAccessUtility** shall be dependant upon network connectivity to the MySQL server.

4.6.12.6. Interfaces:

4.6.12.6.1. None

4.6.12.7. Data:

4.6.12.7.1. User settings; user account information; tasks

5. USER INTERFACE DESIGN

5.1. Overview of User Interface

5.1.1. Several views shall be provided that shall allow the user to both see and control HATA data. The process flows and objects of each major view are demonstrated in this section.

5.2. Screen Images

5.2.1. Original App Prototype (the SRS Modifies this)(See Appendix 22 Diagram 22)

5.3. Adding a Task (See Appendix 7.23 Diagram 23)

5.3.1. Process Flow

5.3.1.1. The user shall select the plus sign in the upper right hand corner of HATA (See Appendix 7.23 Diagram 23 Image 1). The user shall then be shown a list of text input fields in which they may input information about a task, including the task name, task course, task due date, task due time, task type, task reminder, and task notes (See Appendix 7.23 Diagram 23 Image 2). After information is input, the user shall select the save button (See Appendix 7.23 Diagram 23 Image 3).

5.3.2. Objects

5.3.2.1. Object: Input Field (See Appendix 7.24 Diagram 24)

5.3.2.1.1. Description: The input fields shall be where a user shall input information in text form.

5.3.2.1.2. Dimensions: Height: 40 px, Width: 290 px

5.3.2.2. Object: Save Button (See Appendix 7.25 Diagram 25)

5.3.2.2.1. Description: This button shall allow the information input by the user in the input fields to be saved and viewed.

5.3.2.2.2. Dimensions: Height: 65 px, Width: 125 px

5.3.2.3. Object: Advanced Menu Button (See Appendix 7.26 Diagram 26)

5.3.2.3.1. Description: This button shall allow the user to input additional input fields for the task.

5.3.2.3.2. Dimensions: Height: 40 px, Width: 40 px

5.3.2.4. Object: Saving Progress Image (See Appendix 7.27 Diagram 27)

5.3.2.4.1. **Description**: This symbol shall appear after information has been input into the input fields and the save button has been selected.

5.3.2.4.2. **Dimensions**: Height: 180 px, Width: 240 px

5.4. Color Coding (See Appendix 7.28 Diagram 28)

5.4.1. Process Flow

5.4.1.1. The user shall begin in the AssignmentListView of HATA. The user shall be presented with four tabs to sort assignments by: Due Date, Class, Priority, and Completed, which shall sort assignments accordingly, and shall display colors for assignment priority and course. When the customized colors view is selected (the right-most screen in Diagram 28), the user shall be presented with the customized color list, from which they may then use to select colors to assign for color coding of assignments.

5.4.2. Objects

5.4.2.1. Object: Custom Color List Item (See Appendix 7.29 Diagram 29)

5.4.2.1.1. **Description**: An item in the list of customizable colors that shall display the current color for one of 10 custom colors.

5.4.2.1.2. **Dimensions**: Height: 75 px, Width: 320 px

5.4.2.2. Object: Custom Color Picker (See Appendix 7.30 Diagram 30)

5.4.2.2.1. **Description**: A pallet of 10 colors that shall also display the currently selected and previous colors to the left.

5.4.2.2.2. **Dimensions**: Height: 175 px, Width: 300 px

5.4.2.3. Object: Assignment Priority Colors (See Appendix 7.31 Diagram 31)

5.4.2.3.1. **Description**: Three colors that shall represent the priority of an assignment: high, medium, and low. Red shall represent high priority, green shall represent medium priority, and yellow shall represent low priority.

5.4.2.3.2. **Dimensions**: Height: 40 px, Width: 40 px

5.5. Enabling SMS Notifications (See Appendix 7.33 Diagram 33)

5.5.1. Process Flow

5.5.1.1. The user shall select the menu button, select “Settings”, and then the user shall reach the settings view. The user shall then select the sliders

and enable the "SMS Notification Enabled" option. The user shall then select to save the setting.

5.5.2. Objects

5.5.2.1. **Object: Side Menu** (See Appendix 7.34 Diagram 34)

5.5.2.1.1. **Description**: This is where the user shall select the button to gain access to the settings view.

5.5.2.1.2. **Dimensions**: Height: 470px, Width: 200px

5.5.2.2. **Object: Setting Label** (See Appendix 7.35 Diagram 35)

5.5.2.2.1. **Description**: This shall be a description of the setting associated with the label.

5.5.2.2.2. **Dimensions**: Height: 50px, Width: varies to hold the text of the label, Font: Arial, Font Size: 12pt.

5.5.2.3. **Object: Setting Slider** (See Appendix 7.36 Diagram 36)

5.5.2.3.1. **Description**: This shall be used to enable or disable a setting.

5.5.2.3.2. **Dimensions**: Height: 30px, Width: 60px

5.5.2.4. **Object: Save Button** (See Appendix 7.37 Diagram 37)

5.5.2.4.1. **Description**: This button shall be selected by the user to save any changes made to the application settings.

5.5.2.4.2. **Dimensions**: Height: 50px, Width: 75px

5.5.2.5. **Object: Cancel Button** (See Appendix 7.38 Diagram 38)

5.5.2.5.1. **Description**: This button shall be selected by the user to cancel any of the changes they have just made to the application settings.

5.5.2.5.2. **Dimensions**: Height: 50px, Width: 75px

5.6. **Calendar View** (See Appendix 7.39 Diagram 39)

5.6.1. Process Flow

5.6.1.1. The user shall see the monthly calendar view by default. The user shall then select the weekly view tab from the list of tabs displayed at the top of the default view to see the weekly view, and the daily view tab to see the daily view.

5.6.2. Objects

5.6.2.1. **Object: Monthly Calendar View** (See Appendix 7.40 Diagram 40)

5.6.2.1.1. **Description**: This view shall display tasks for the whole month.

5.6.2.1.2. **Dimensions**: Height: 390 px, Width: 390 px, Font: Arial, Font Size: 12pt.

5.6.2.2. Object: Weekly Calendar View (See Appendix 7.41 Diagram 41)

5.6.2.2.1. **Description**: This view shall display tasks for the whole week.

5.6.2.2.2. **Dimensions**: Height: 470 px, Width: 390 px, Font: Arial, Font Size: 12pt.

5.6.2.3. Object: Daily Calendar View (See Appendix 7.42 Diagram 42)

5.6.2.3.1. **Description**: This view shall display tasks for the whole day.

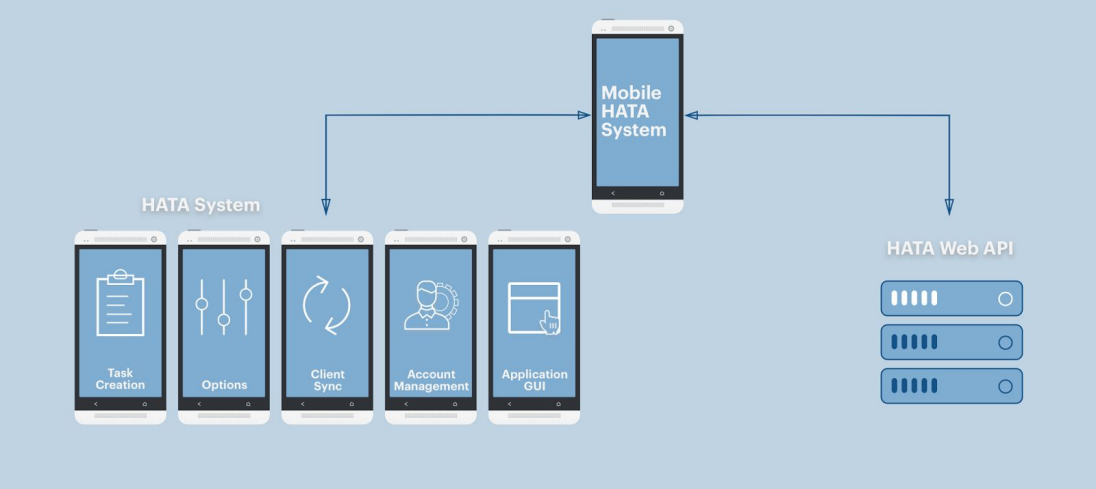
5.6.2.3.2. **Dimensions**: Height: 500 px(Scrollable), Width: 390 px, Font: Arial, Font Size: 12pt.

6. REQUIREMENTS MATRIX

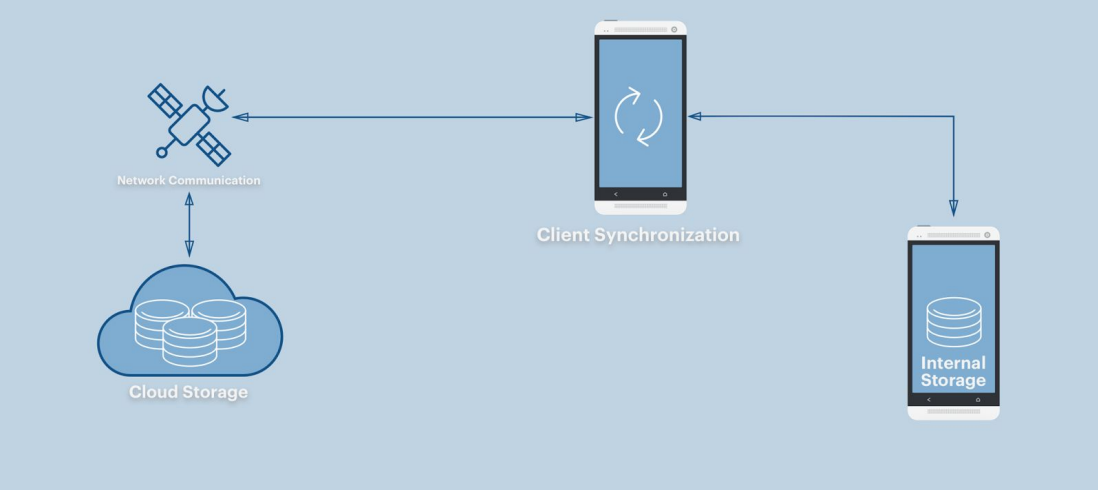
6.1. View and edit the Traceability Matrix here.

7. APPENDIX

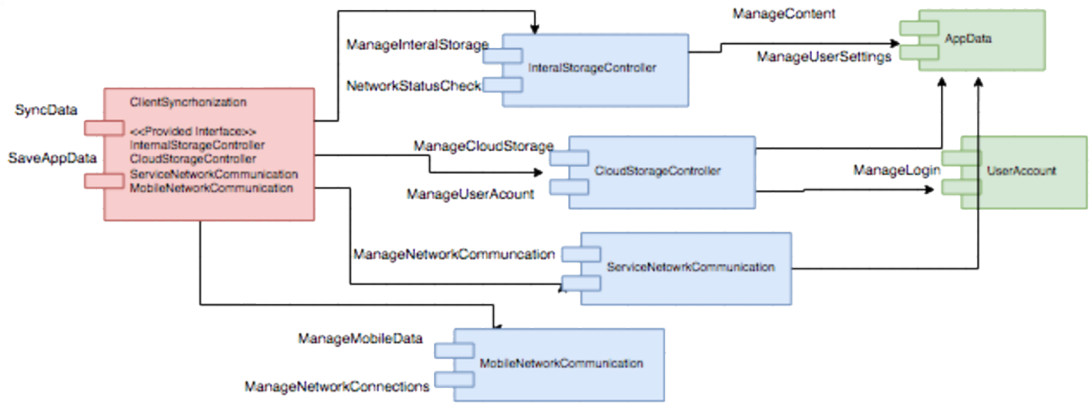
7.1. Diagram 1 Mobile HATA Relationship - Demonstrates the relationship between the Mobile HATA and the Mobile HATA System. The Mobile HATA can be broken down into Task Creation, Options, Client Synchronization, Account Management, and Application GUI.



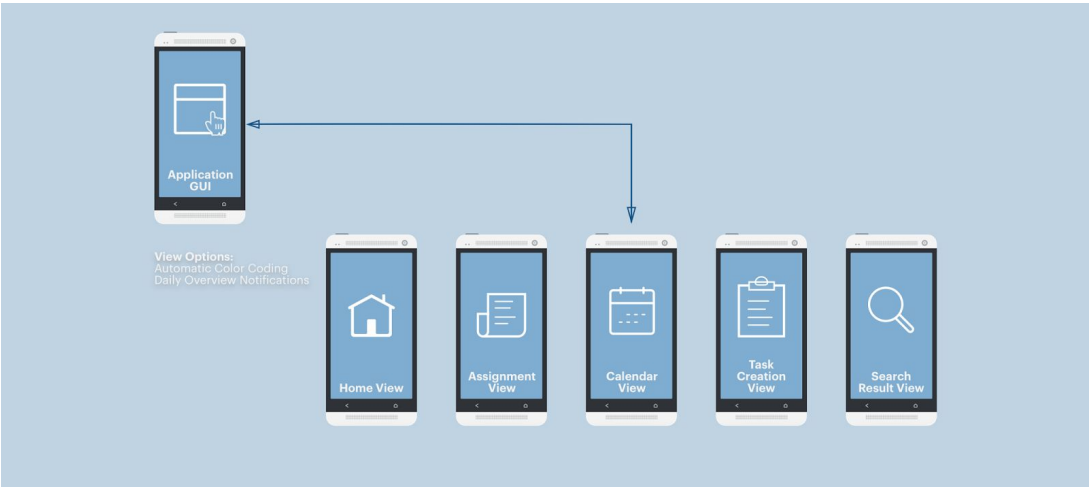
7.2. Diagram 2 Client Synchronization Data Sync - Breaks down the Client Synchronization and how it syncs data from internal storage and cloud storage.



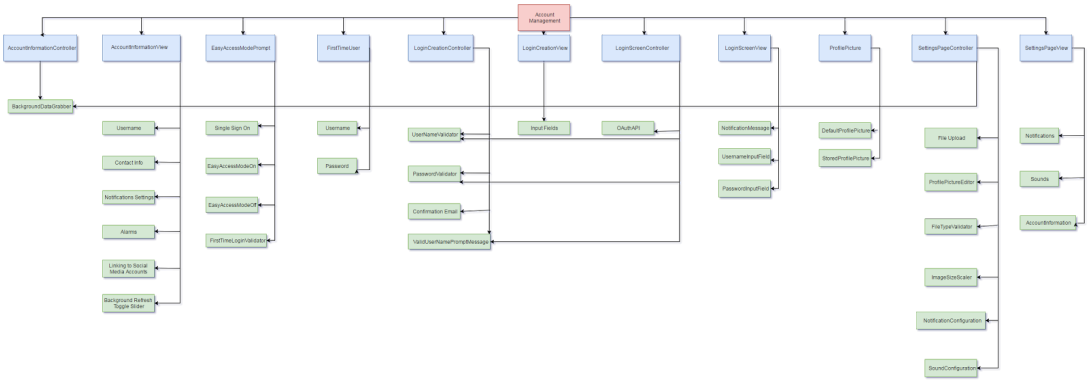
7.3. Diagram 3 Component UML Diagram - Diagram for Client Synchronization and shows how various parts of the system communicate.



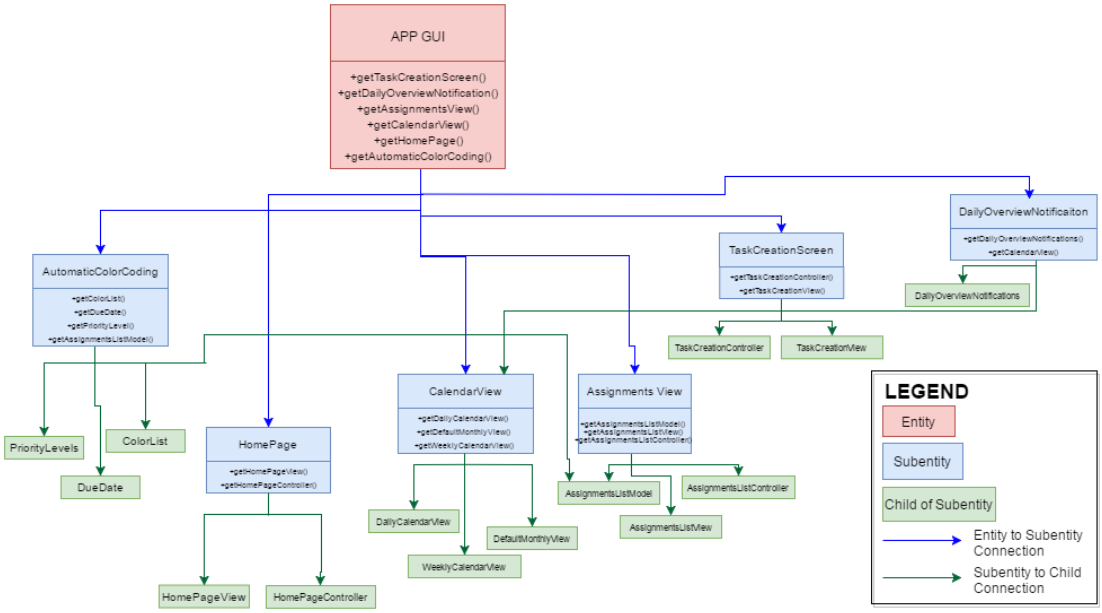
7.4. Diagram 4 Application GUI Views - Breaks down the Application GUI into the different views that will be displayed to the user.



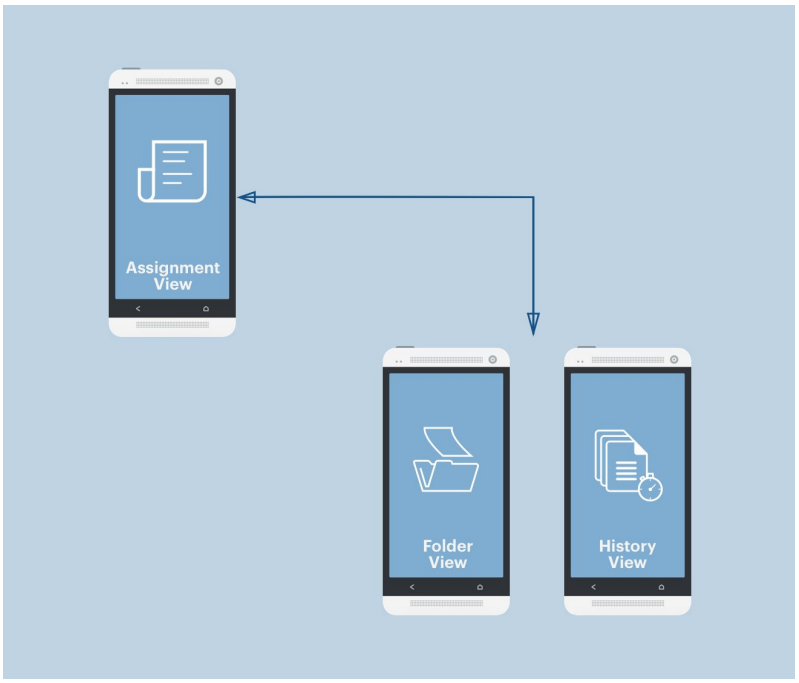
7.5. Diagram 5 Account Management - Shows how Account Management interfaces with other classes and how they interface with each other. Use of the Account Management will call at least one of its entities, which may also require data from another entity.



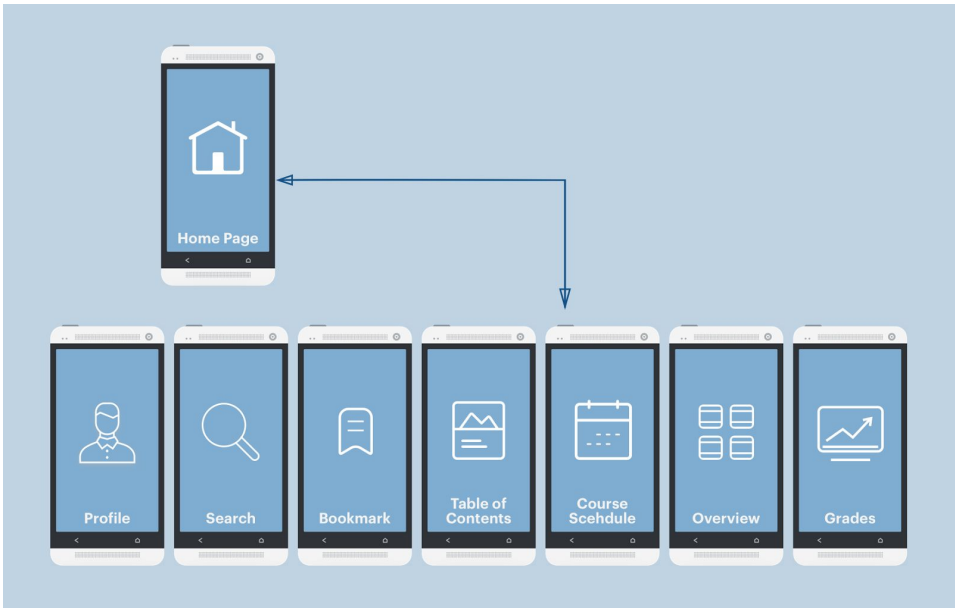
7.6. Diagram 6 App GUI - Shows how App Gui interfaces with other classes and how they interface with each other. Use of the App Gui will call at least one of its entities, which may also require data from another entity.



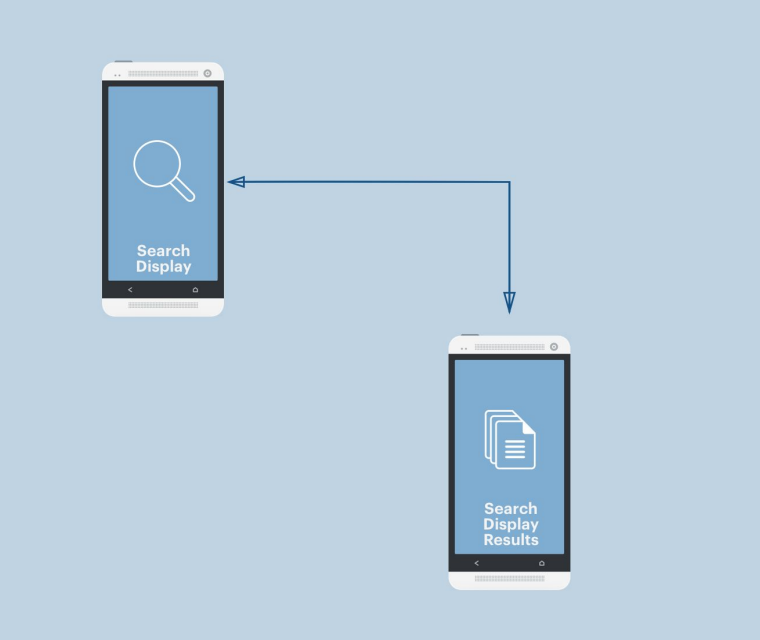
7.7. **Diagram 7 Assignment View -** Shows how the assignment view can lead into the folder view or into the assignment history view.



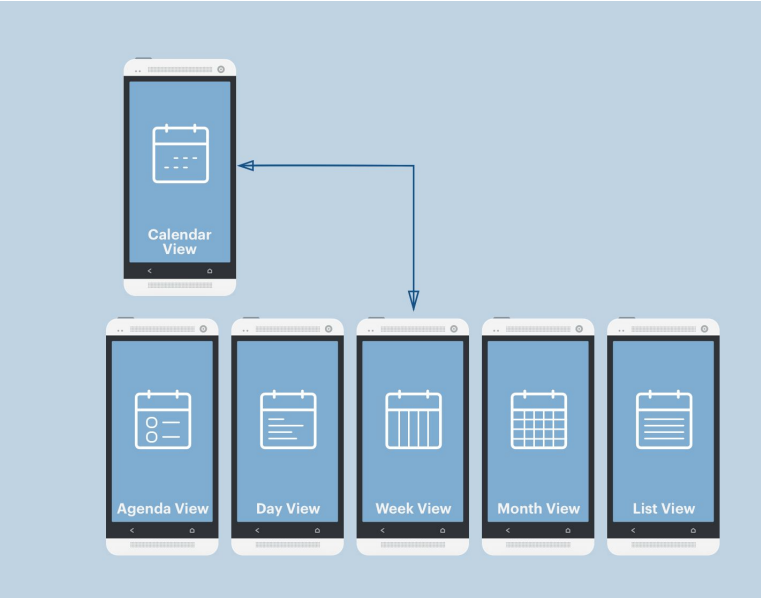
7.8. **Diagram 8 Home Page View -** Shows how the Home Page can lead to many other views and options. On all views you will have the option of selecting any other view, hence the bi-diagonal arrows.



7.9. **Diagram 9 Search Page View -** Shows that when search is used on the HATA Mobile Application, the user will be taken to a search results screen. The user can return back to the screen the Search Display component is on from the Search Display Results.

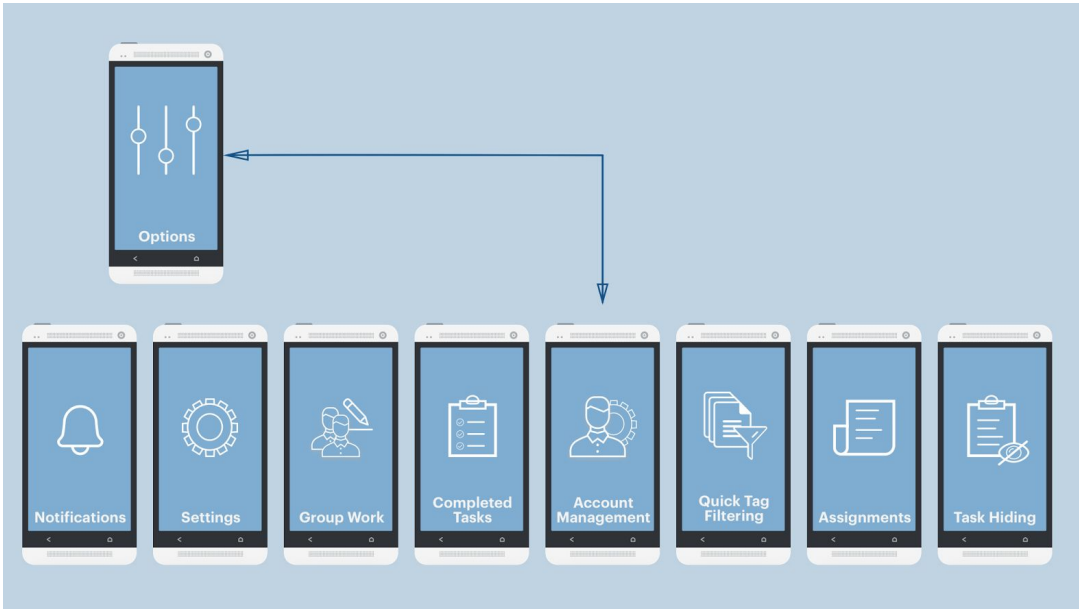


7.10. **Diagram 10 Calendar Page View -** Shows the different ways that the calendar view can be organized.

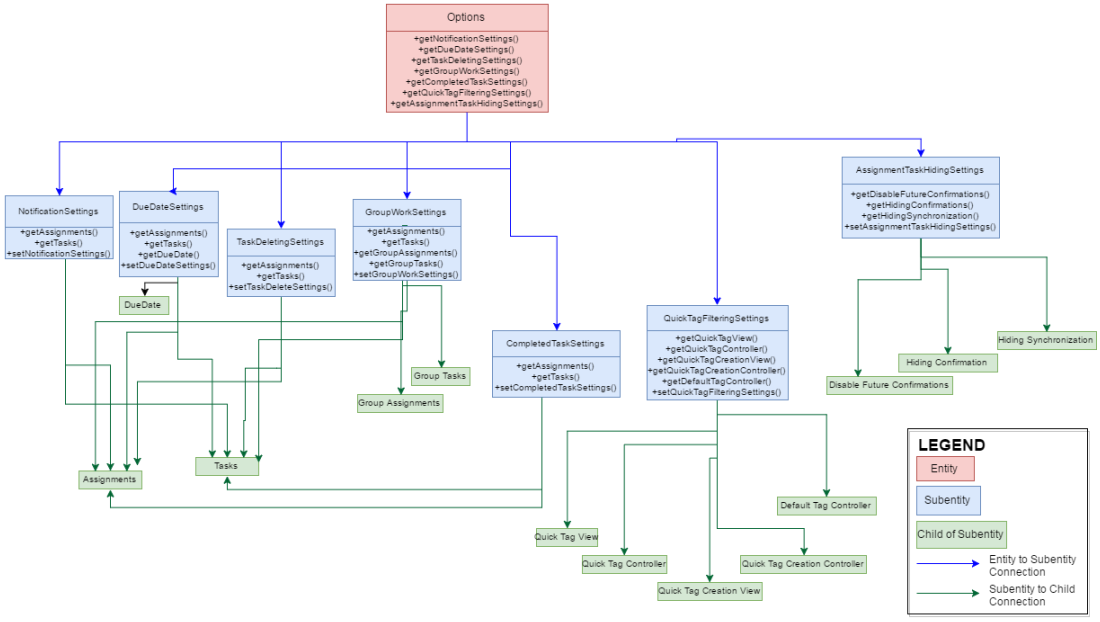


7.11. Diagram 11 Task Creation Screen - Shows each editable field that the user can input data into from the Task Creation Screen that results in a new task.

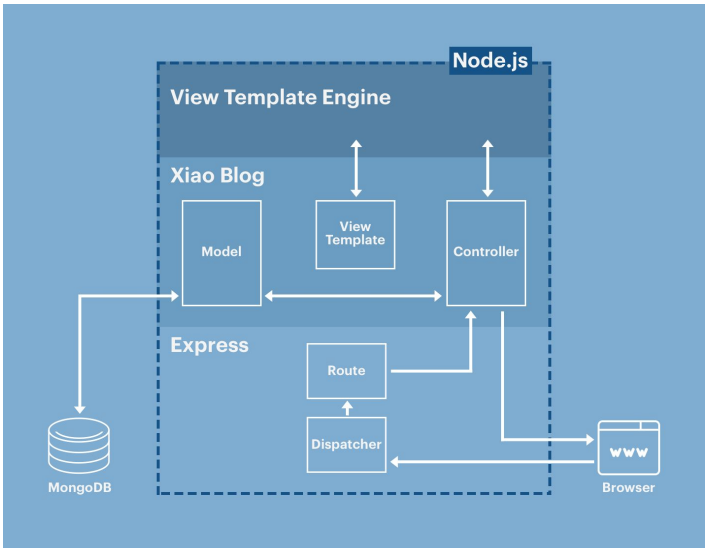
7.12. Diagram 12 Option View - Lists out the different settings that can be found in the options menu.



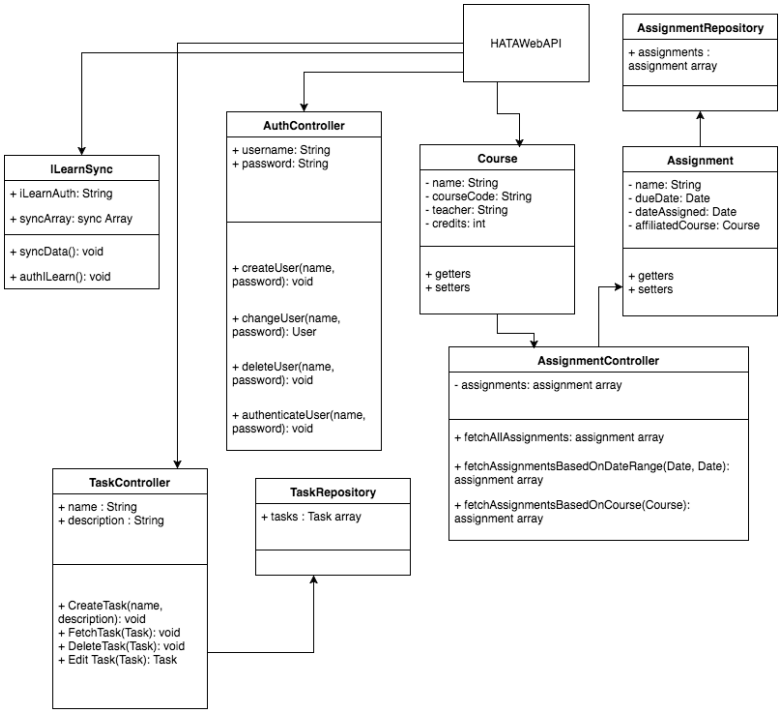
7.13. Diagram 13 Option UML view - Is a class UML diagram that shows that Options is composed of seven settings: NotificationSettings, DueDateSettings, TaskDeletingSettings, GroupWorkSettings, CompletedTaskSettings, QuickTagFilteringSettings, and AssignmentTaskSettings. Each of these settings utilize data layer classes and some controller classes from AppGUI.



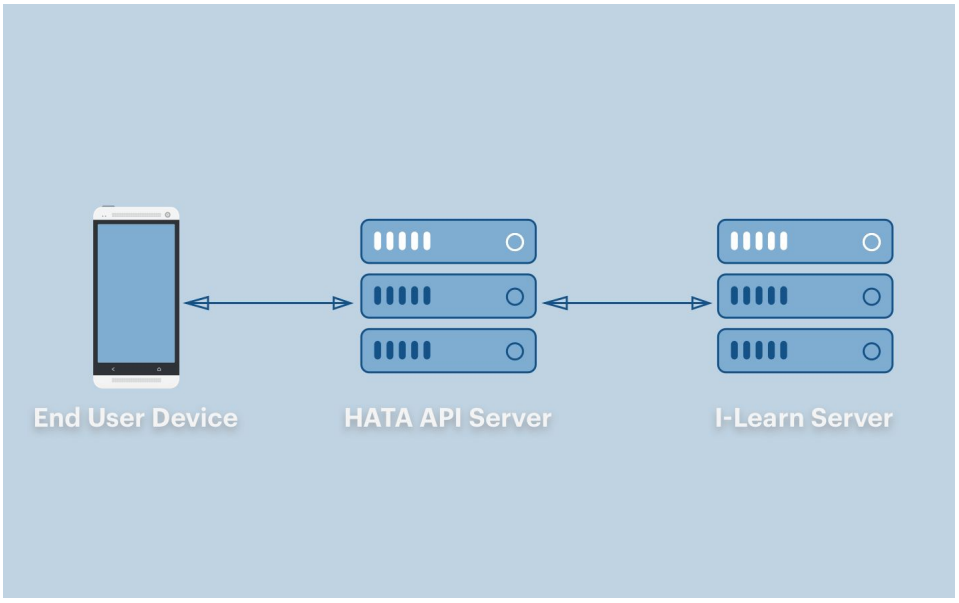
7.14. Diagram 14 Framework View [4] - Demonstrates how Express Framework works.



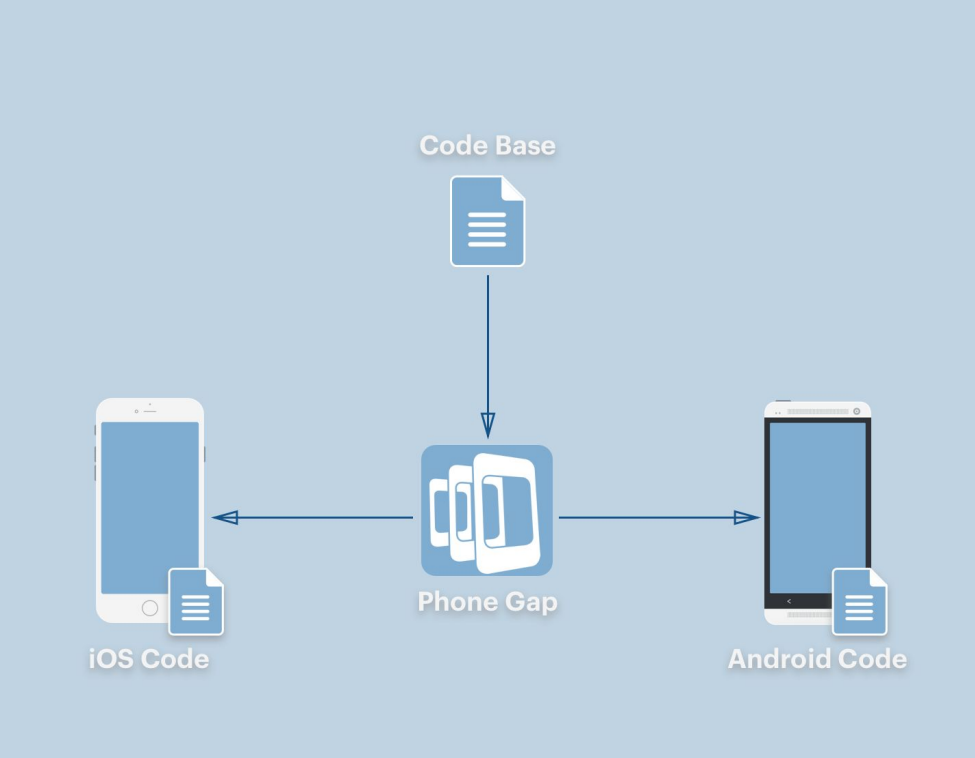
7.15. Diagram 15 HATAWebAPI UML Class Diagram - This UML class diagram demonstrates how the entities of the HATAWebAPI are related to each other.



7.16. Diagram 16 HATAWebAPI - HATA Client communicates to HATAWebAPI to get a list of assignments and tasks. HATAWebAPI talks back and forth to I-Learn to receive assignment lists. Merging is done on HATAWebAPI.



7.17. Diagram 17 Code Base View - Client application has a single code base that is processed by PhoneGap. This processing yields the artifacts necessary for deployment on the Android and IOS platforms.

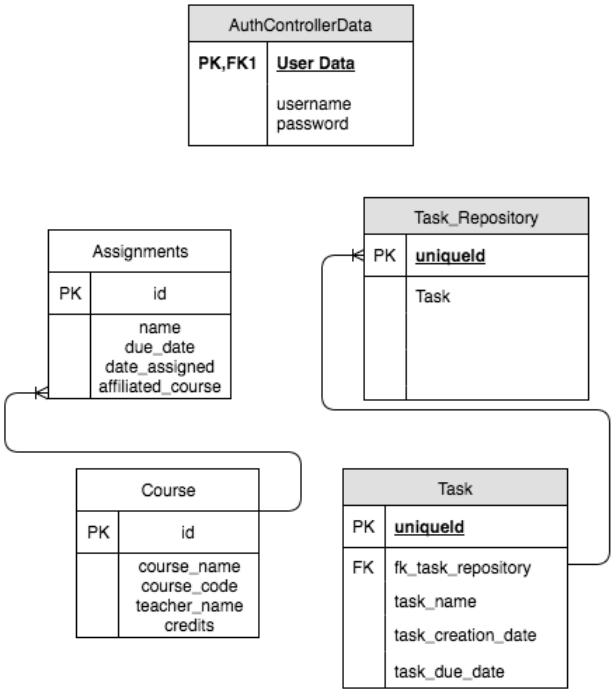


7.18. Diagram 18: Removed from document and left for reference.

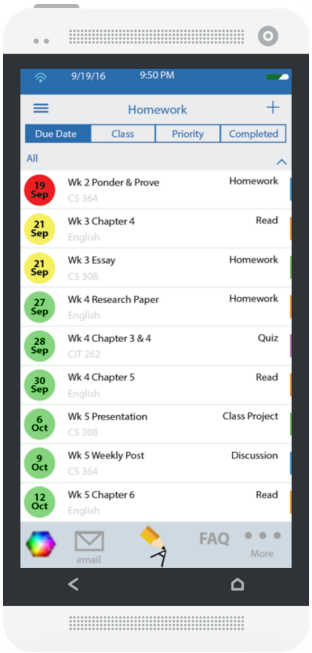
7.19. Diagram 19: Removed from document and left for reference.

7.20. Diagram 20:Removed from document and left for reference.

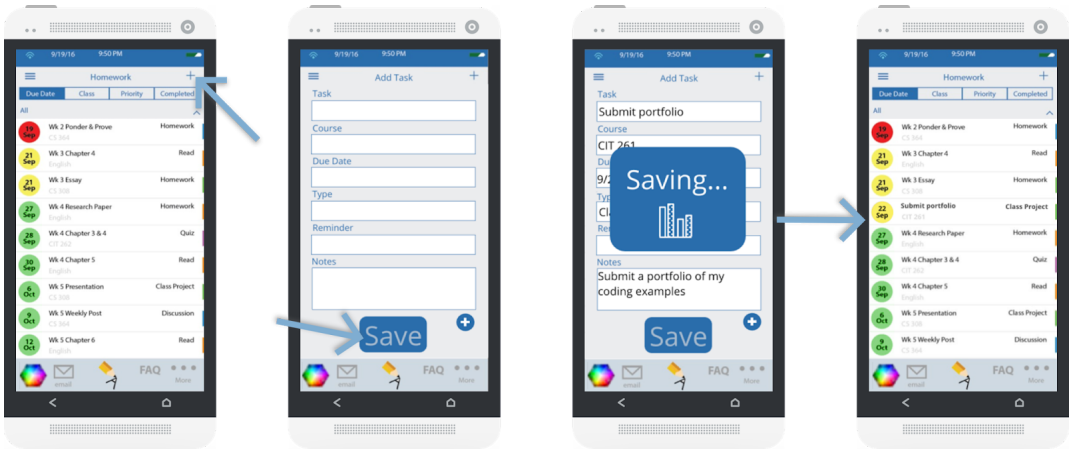
7.21. Diagram 21 HATAWebAPI Entity Relationship Diagram: A diagram to show the relationships between the tables in the relational database of the HATAWebAPI.



7.22. Diagram 22: A screenshot of HATA application prototype. This shows the Due Date View ordering assignment and tasks by their reminder time.



7.23. Diagram 23: Adding a task: Screenshots of the flow of adding a task and the steps to take.



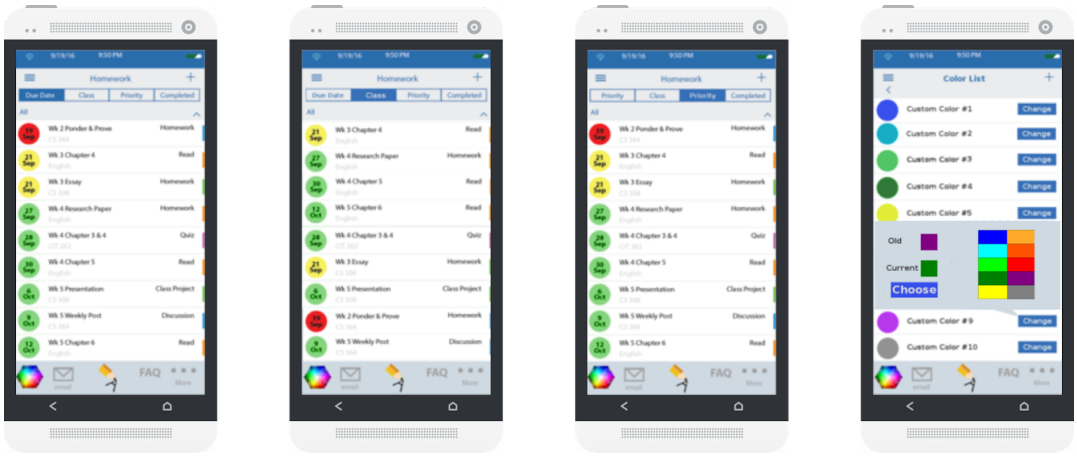
7.24. Diagram 24: Input Field: A text field where the user shall input information about a task.

7.25. Diagram 25: Save Button: A button that shall be selected in order to store the task information as a task.

7.26. Diagram 26: Advanced Menu Button: This button shall allow a user to create additional details about a task when selected.

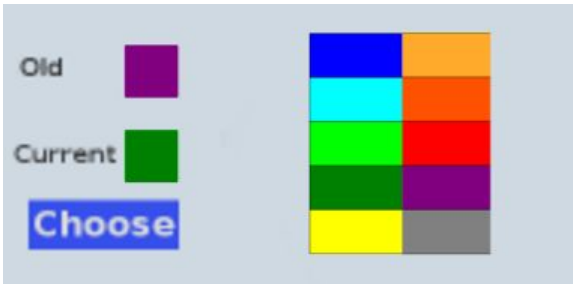
7.27. **Diagram 27: Saving Progress Image:** This image will be shown to let the user know HATA is in the progress of saving a task.

7.28. **Diagram 28: Color Coding**: Screenshots of the process flow of the different assignment sorting options and their effect on an assignment list, as well as the custom colors screen.



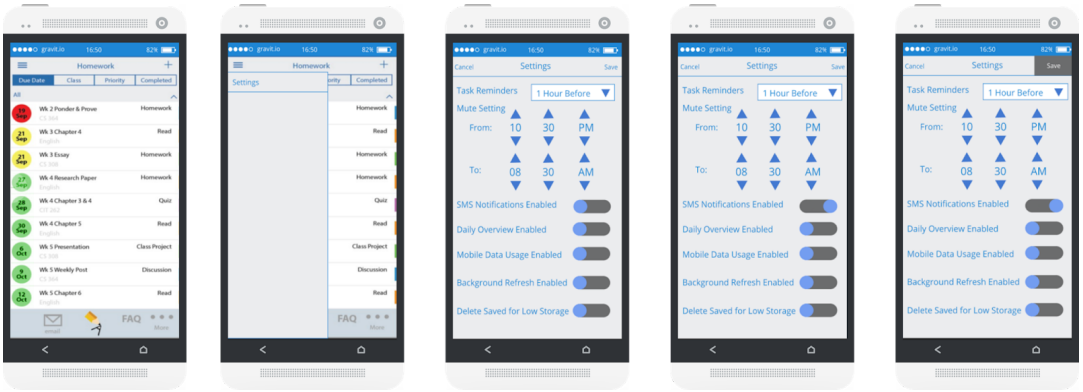
7.29. **Diagram 29: Custom Color List Item**: A single element in the list of customizable colors. The circle on the left represents the current color for custom color #3.

7.30. Diagram 30: Custom Color Picker: A color picker that appears after selecting the 'change' button on a custom color list item. On the left, it displays the previously selected color, as well as the new color. On the right, it displays a palette of 10 colors from which the student can select custom colors for tasks and assignments.

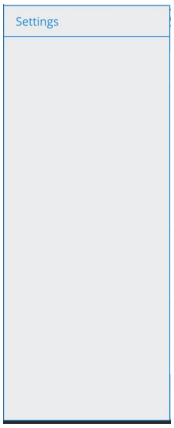


7.31. Diagram 31: Assignment Priority Colors: The colors representing priority displayed to the left of assignments in an assignment list.

7.32. Diagram 33: SMS Notifications Enabled : This shall be the process for enabling SMS notifications in the HATA mobile app.



7.33. Diagram 34: Side Menu - This is where the user shall select the button to gain access to the settings view.

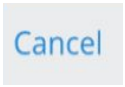


7.34. Diagram 35: Setting Label - This shall be a description of the setting associated with the label**.**

7.35. Diagram 36: Setting Slider - This shall be used to enable or disable a setting.

7.36. Diagram 37: Save Button - This button shall be selected by the user to save any changes made to the application settings.

7.37. Diagram 38: Cancel Button - This button shall be selected by the user to cancel any of the changes they have just made to the application settings.



7.38. Diagram 39: Calendar View - This view shows the different calendar views as they will appear to the user.

7.39. Diagram 40: Monthly Calendar View - This view shows the way the HATA will present a month long period to the user.

January 2015						
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
				1 New Years Day	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19 M L King DaY	20	21	22	23	24
25	26	27	28	29	30	31

7.40. Diagram 41: Weekly Calendar View - This view shows the way the HATA will present a week long period to the user.

December 27 Sunday	
December 28 Monday	
December 29 Tuesday	
December 30 Wednesday	
December 31 Thursday	
January 1 Friday	New Year's Day
January 2 Saturday	

7.41. Diagram 42: Daily Calendar View - This view shows the way the HATA will present a one day long period to the user.

07:00 AM	
08:00 AM	
09:00 AM	
10:00 AM	
11:00 AM	
12:00 PM	
01:00 PM	
02:00 PM	
03:00 PM	
04:00 PM	
05:00 PM	
06:00 PM	