

**Course ID: CPS5995**

**Project - T.A. Solutions**

**Software Design Document (SDD)**

**Version 1.0**

**Prepared by: Andres, Robert, Matt, Kyle**

**Reviewed by : Brian, Chris**

**Instructor : Dr. Jing-Chiou Liou**

**Date:02/16/2020**

Table of Contents

1 . Introduction………………………..........................................................................……...3

1.1 Purpose…………………..................................................................................3

1.2 Scope……………………….............................................................................3

1.3 Goals and Objectives…………….....................................................................4

1.4 Acronyms and Abbreviations.....…………………………...............................4

Definitions and Acronyms.......................................................................................4

2. System Design………………………………………….....................................................4

2.1 System Architecture…………………………………………….................. ....4

2.2 Context Diagram…………………………………............................................5

2.3 Use case Diagram..............................................................................................6

3. System Functional Model……………………………………………………..................10

3.1 Representation of functional Modules ………………………………............11

3.2 Software Process Model ……………….........................................................11

4. Data Design……………………..……………………..............…....................................13

4.1 Data Base Schema Diagram……………........................................................13

5. Design Constraints, Restrictions and Limitations……………………............................14

5.1 Design Constraints ..........................................................................................14

5.2 System Restrictions and Limitations ..............................................................14

6. Tools and References.........................................................................................................15

6.1 Tools used to create Diagrams ........................................................................15

6.2 Documentation Tool........................................................................................15

6.3 Reference Materials.........................................................................................15

**Add version history**

**Fix spacing to reflect how it looks in word**

Check for any changes from original not documented

**1. Introduction**

Currently, Code Samurai (the Computer Science tutoring program at Kean University), uses Google Sheets as a check-in/out method for Students and Tutors. Prof. Wang, the Program Coordinator, has explained to the tutors that it is important to sign each student in for tutoring because it will determine whether more tutors are needed in these subjects or time frames. However, often these logs are not filled out because tutors get caught up helping and forget to check-in/out. Additionally, tutors often have difficulty remembering to clock-in and clock-out both by paper and using the google document because of personal time constraints (rushing to class after the shift) or system failures (loss of wifi internet connectivity). Because of these issues, the Code Samurai program does not have the data it needs to make decisions about the future of the program including what courses to prioritize, when is the service is most utilized, and more.

## **1.1 Purpose**

Create web and mobile applications that will streamline the check-in/out process for tutors and students as well as log all their activity to provide the administration with analytics of the tutoring and subjects. The benefit of this application is that it can be used for programs other than code samurai. All of the data will be stored in a centralized database where it can be used for larger-scope analytics and machine learning tools. For the central end-users (students and tutors, for example), this application serves as a convenient means to check-in, clock-in, and network with other students and explore other programs. However, the organization controlling this software can use our machine learning and analytics tools to develop new knowledge based on individual programs are all programs together.

## **1.2 Scope**

By the end of the timeline there should be a successful development of the following:

Mobile App

Web application

Analytics and reports

Data processing/Analytic Tools

There should be successfully tested the system by either deploying it to code samurai or a

small tutoring group.

## **1.3 Goals and Objectives**

* Signup and Login features will be fully implemented.
* Clock-In, Clock-Out, Check-In, and Check-out functionality will be fully implemented.
* Tutor Schedule, Tutor Rating features will be fully implemented.
* Data Analytics and Machine Learning Tools functionality will be fully implemented.

## **1.4 Acronyms and Abbreviations**

* SDD - Software Design Document
* UI - User Interface
* PHP - Personal Home Page, a server-side scripting Language
* HTML - HyperText Markup Language, for web browser UI
* CSS - Cascading Style Sheets, for UI

## **2. System Design**

## **2.1 System Architecture**

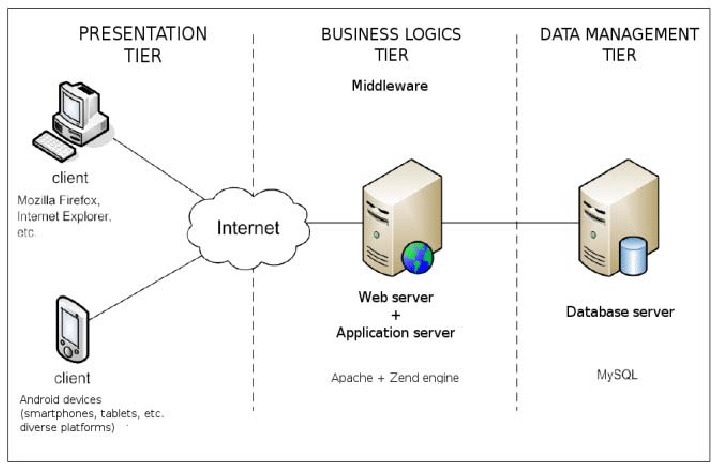
The T.A. Solutions application will consist in a 3-tier architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules as follows.

Presentation Layer: it will display the content to students on their cell phones or computer browsers in the form of an application.

Application Layer: the middle layer of the model, it assures the calculations and operations performed between input-output requirements and data. Also known as the application server.

Data Layer: it is the database management system that will provide access to the application data.

Below it is a high level overview of a 3-tier architecture.



**Figure 1 Three Tier Architecture**

**2.2 Context diagram :**

This diagram represents the entities which interact with the The T.A. Solution System.

Admin

Supervisor

Tutor

Student

**Figure?**

**Modify this to reflect 1:Many and M:N**

Admin : Admin can create and maintain the Supervisors’ accounts. Also, there is only one admin account whose credentials will be shared with the system support people(via document) and therefore no user interface provided to alter the credentials of this account which can be done only by the system support people via the back end. Admin is also the only one that can create programs.

Supervisor: Supervisor accounts must be created by the Admin. Supervisors can assign users “tutor roles” allowing them to act as tutors for a particular program. Similarly, they can remove tutor-roles from users as well. Supervisors have access to all tutor data related to their programs. Supervisors can view graphic visualizations of that data, make use of data analytics features and machine learning features built to aid their decision making regarding the program.

Tutor: A Tutor must be provided a tutor role by a Supervisor for a particular program. A user can have multiple tutor roles, each one with a different program. As a tutor, a user can clock-in and out, keep a public schedule, check-in and check-out students, as well as view any of their personal data. They have some data visualization features, as well as data analytics and machine learning features that will help them improve and prepare for the future. There are also community features that allow tutors to interact with each other.

Student: A student role is the default role everyone is provided when creating an account. This allows them to check-in and out with a tutor, as well as explore what programs are available and what tutors best fulfill their needs. A student role only controls his or her own data.

**2.3 Use case Diagram**

The use case diagram or usually referred as behavior diagram is used to describe a set of actions “use cases” that our T.A. Solution system “subject” can perform in collaboration with one or more external users of the system called “actors”. The use cases below are represented by eclipse and the software system is represented by a rectangle.

**Reflect changes from Requirements Doc for UML**

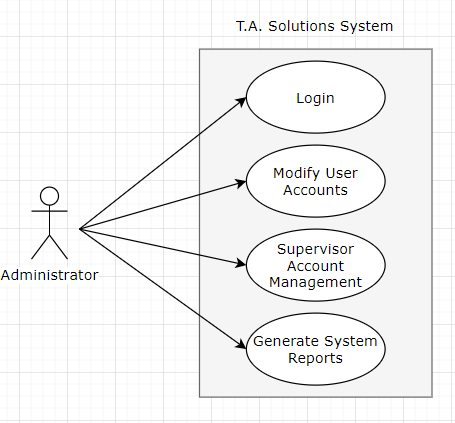


Figure 2 : Use case diagram for Admin

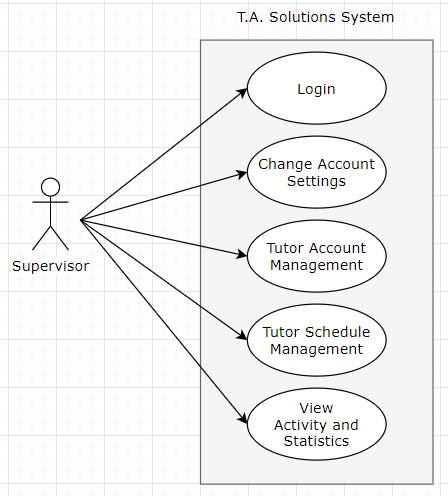


Figure 3: Use case diagram for Supervisor

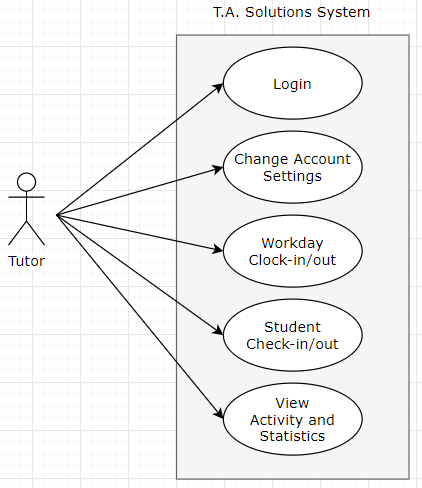


Figure 4: Use case diagram for Tutor

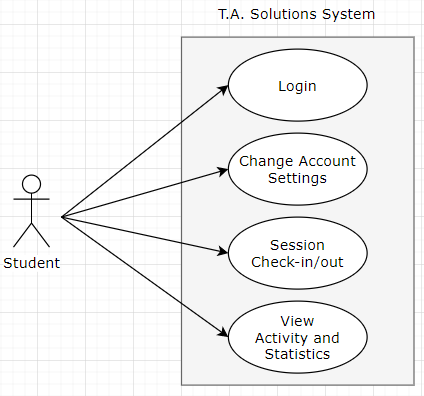


Figure 5: Use case diagram for Student

**3. System Functional Model**

**3.1 Representation of functional Modules:**

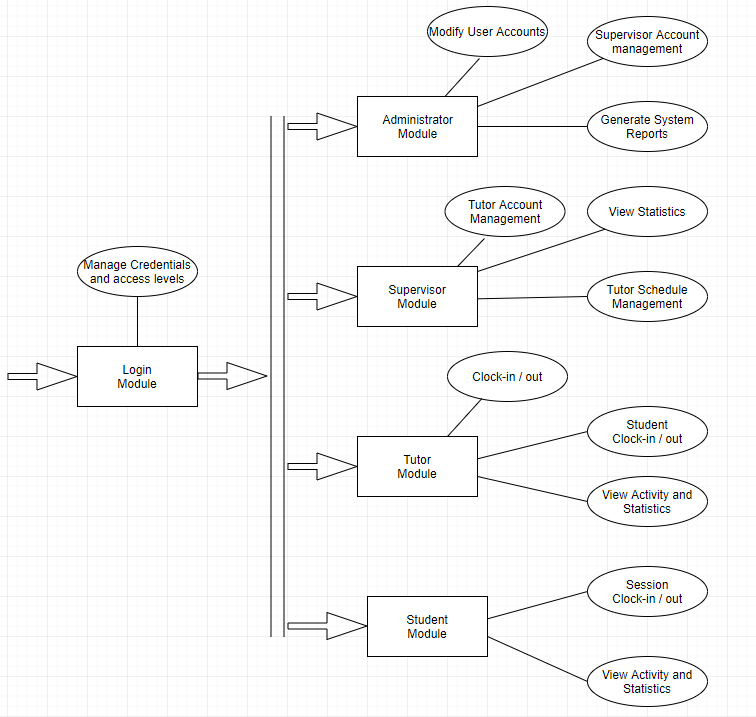
****

Figure 6: Use case diagram for Student

**Merge common parts across modules**

**3.2 Software Process Model**

The waterfall approach is applied to develop this software system as the main focus is to develop and deploy parts at different times. This method allows for the software to be put into use early and functions can be added afterwards without changing the deployed software functions.

The waterfall model is also known as a linear-sequential life cycle where the next phase relies on the previous phase as input throughout the software development cycle. Starting with the requirements phase, each phase that follows what was setup in the previous phase as requirements. This allows for an earlier integration test such that a prototype app can be deployed to collect data while the data visualization function is designed. The early phases of development allow for client feedback before coding starts as well as minor feedback after each phase. Since all phases were planned out from the beginning, each integration test should not require much changes as the output of the previous phase was known.

## Below is a visualization of the waterfall model.

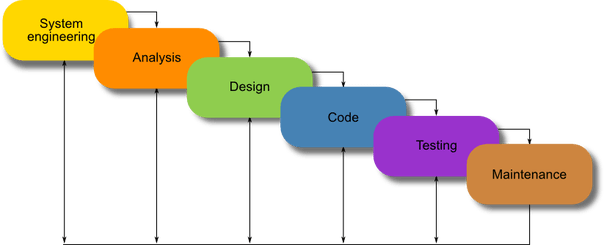


Figure 7: Waterfall - Software Process Model

This software system is setup to be developed in three phases.

· Phase 1:

o Staging Phase

o Basic app with sign-up, sign-in, portals for user type

o Logs user data to database

· Phase 2:

o Test Case and Test Plan

o Student/Tutor Check-in System

o Code Samurai Requirements

· Phase 3:

o Administrative functions: adding/removing tutors, adding/removing tutor subjects, view history

o Tutoring analytics

**4. Data Design**

**4.1 Database Schema Diagram :**

A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams.

Mysql Database is used in this system. All Database tables are structured to conform to third normal form to ensure a better database structure.

The Database schema diagram represented here is based on the initial design of the software and therefore might be changed/updated in forth-coming iterations.

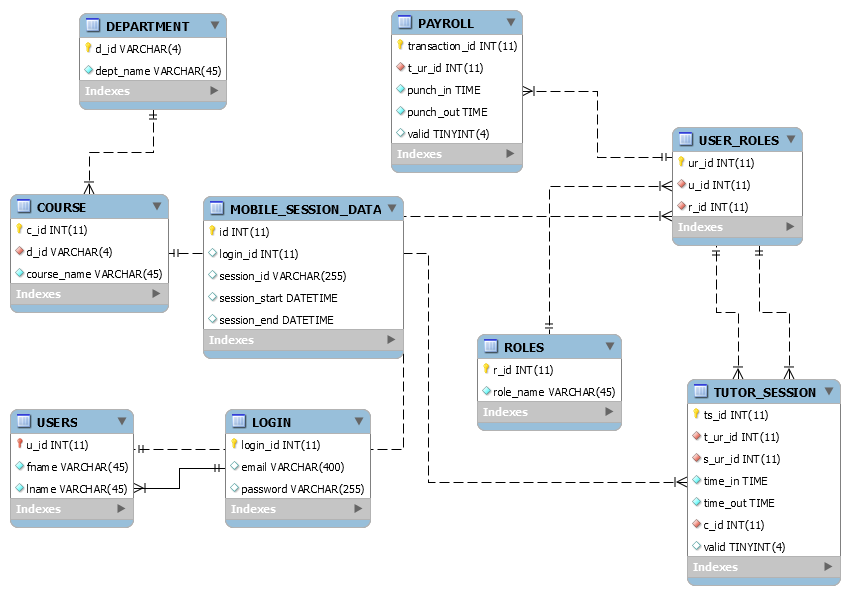
****

Figure 8: Database Schema Diagram of T.A. Solution System

## **5. Design constraints, Restrictions, limitations**

**5.1 Design Constraints :**

The T.A. Solutions system is a short time project of only three months. The system is designed in two weeks to facilitate the team with the basic sketch of the intended software system.A

## **5.2 System Restrictions and Limitations**

T.A. Solutions system is a web and mobile application therefore it requires internet access or a mobile application installed on the student’s cell phone. Also a web and database server must be available to access and store the data.

This System is meant to people with basic knowledge on computer usage.

**6. Tools and References**

**6.1 Tools used to create Diagrams**

* Database schema Diagram is generated using Mysql workbench.
* Use Case Diagrams have been generated by the Online tool from the web “www.creately.com”

**6.2 Documentation Tool**

This document has been created using Google Docs and Microsoft Office 2016.

**6.3 Reference Materials**

* <https://www.uml-diagrams.org/examples/android-uml-examples.html>
* <https://www.draw.io/>
* <https://www.ibm.com/automation/process-mapping?p1=Search>
* <https://dev.mysql.com/doc/workbench/en/wb-creating-eer-diagram.html>
* <https://stackoverflow.com/questions/2488/auto-generate-database-diagram-mysql>