SMART

QuickManage
Software Architecture Document

Version <1.0>

Revision History

Date	Version	Description	Author
<dd/mmm/yy>	<x.x>	<details>	<name>

Table of Contents

Software Architecture Document

1. Introduction

1.1 Purpose

- This document provides an architectural overview of the QuickManage system. There are some feature of the QuickManage system which are described by using some different architectural view .It is built to analysis and impart the important architectural decisions.

1.2 Scope

- This document have been created by SMART and implemented in RUP model. The generality of the sections have been taken from RUP model combine with the Software Architecture Document template.
- This document also provides an overview of the architecture of QuickManage system. This program intends to be used in the music school.

1.3 References

- http://www.agilemodeling.com/essays/agileModelingRUP.htm
- Priestley, M. (2004). *Practical Object-Oriented Design with UML.* Mc Graw Hill.

2. Architectural Goals and Constraints

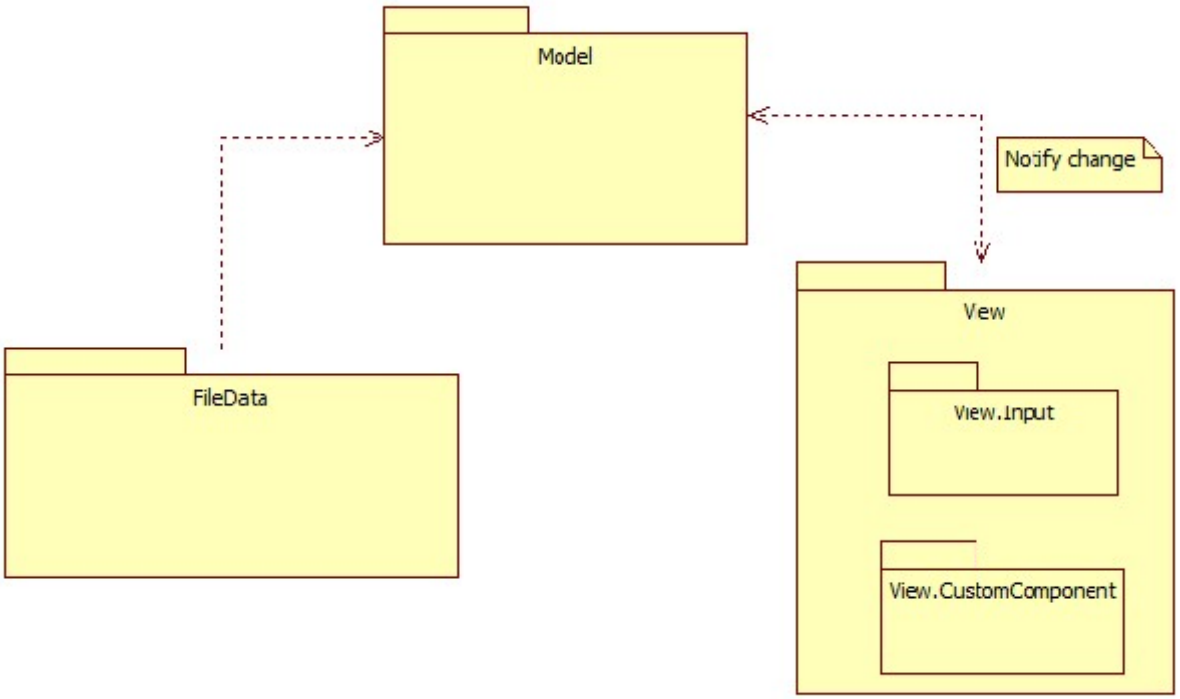
- MVC must be used in the system's design.
- The software development process must follow the RUP Model.
- Abstract class, inheritance, polymorphism, interface (the principle of OO concepts) must be followed.
- Portability: This program is developed in Java – a portable programming language that can run at any computer, on any platform that has Java Runtime Environment. Data of the program is also exportable.
- Distribution: None.
- Reuse: Divide and implements many interfaces and abstract classes.
- Development tools: Netbeans 7.2, JDK 7.0, Subversion 1.7 with Unfuddle.com SVN, Microsoft Word 2010.
- Team structure: 4members: 1 leader, 2 coders, 1 diagram drawer.
- Schedule: 3 weeks and 3 days. The 1st week is used for planning and researching. The 2nd week is used to coding. The 3rd week is used to check, fix and patch bugs. The remaining is used to finalize everything in QuickManage project.

3. Logical View

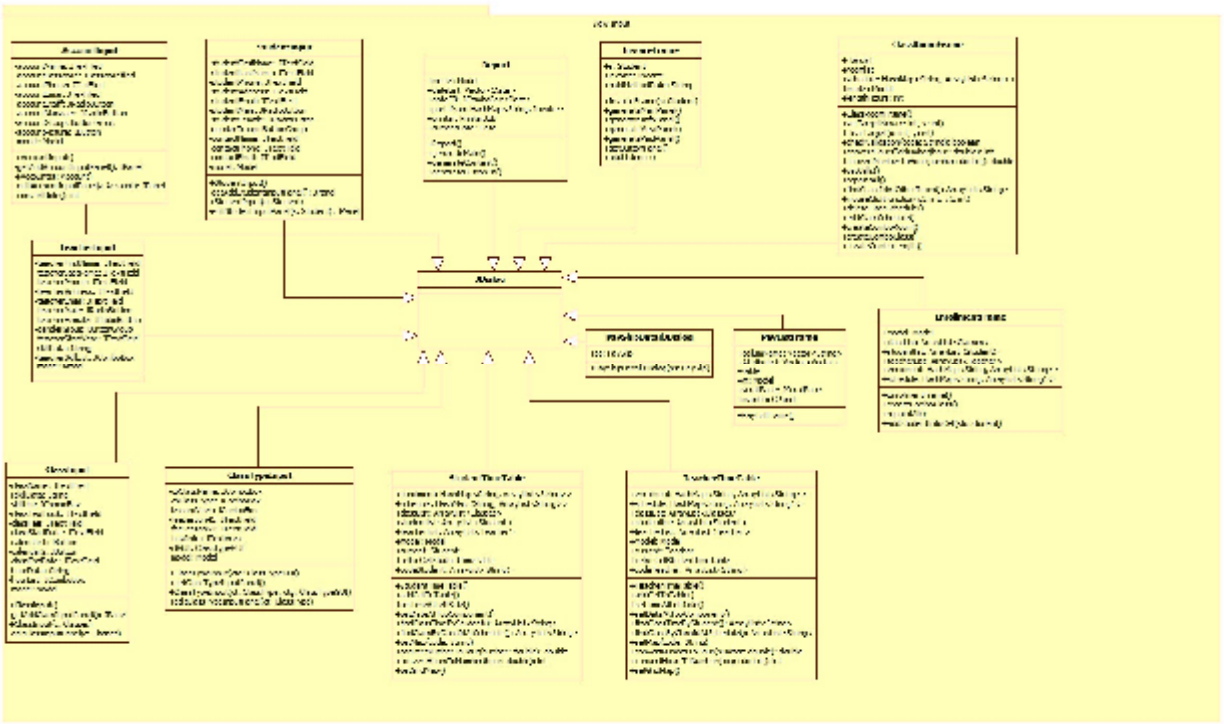
3.1 Overview

- The system will be designed based on the MVC model.
- In Model package (include Controller manager), it comprises the main classes such as: Account, Student, Class, and Teacher. Moreover, it can help the system to manage data, functions in the logic way.
- In View package, it helps the system to create an easy understanding user interface.

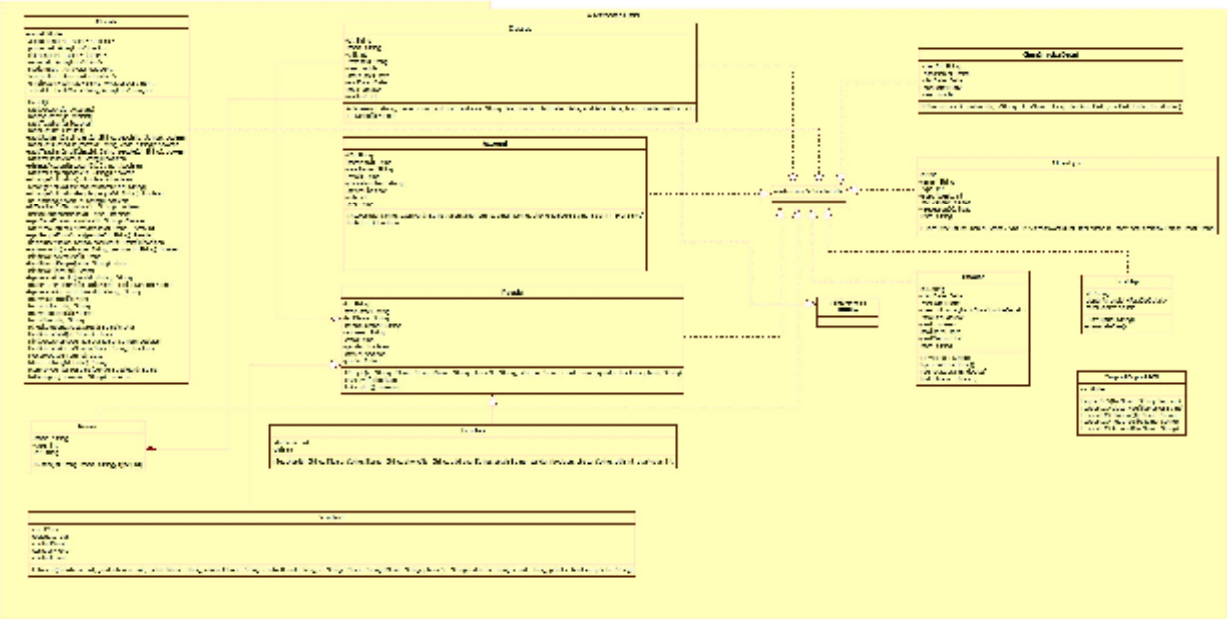
3.2 Package and Class Diagrams



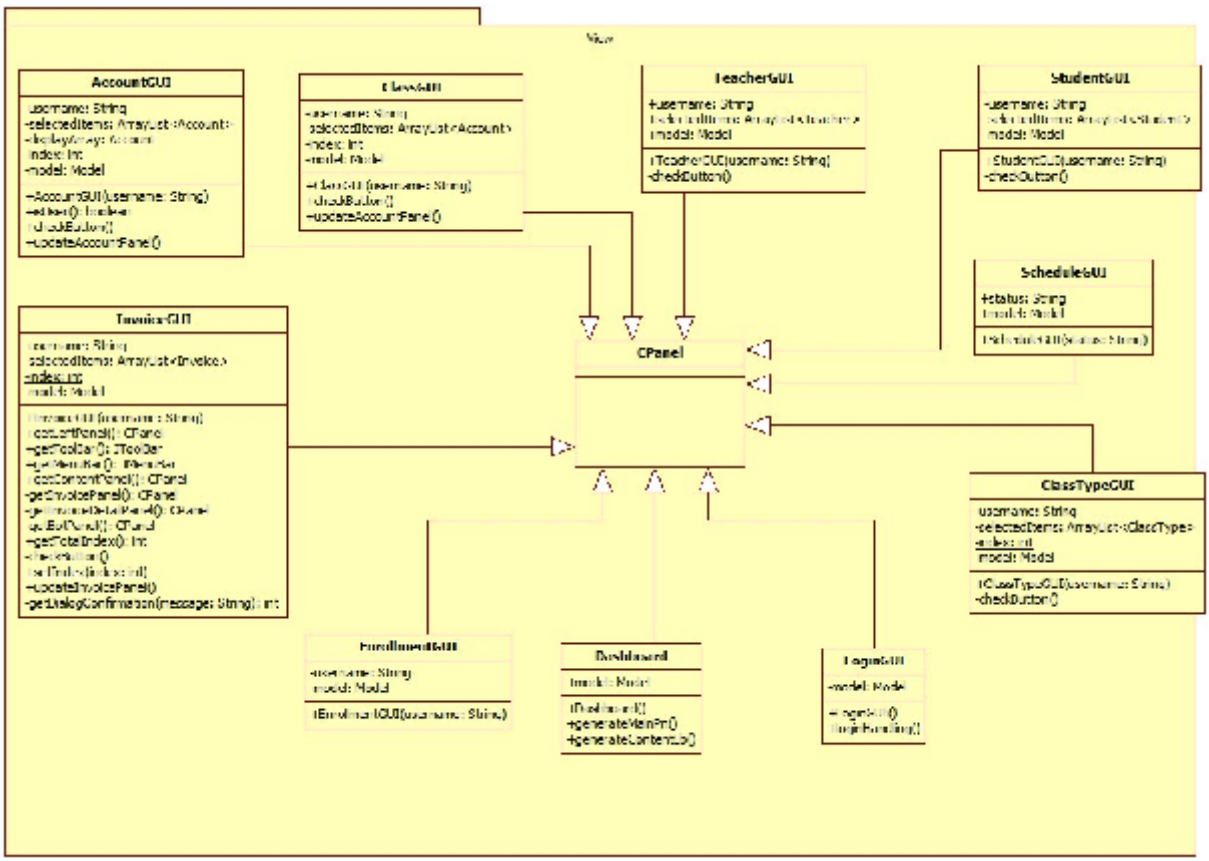
QUICKMANAGE PACKAGE



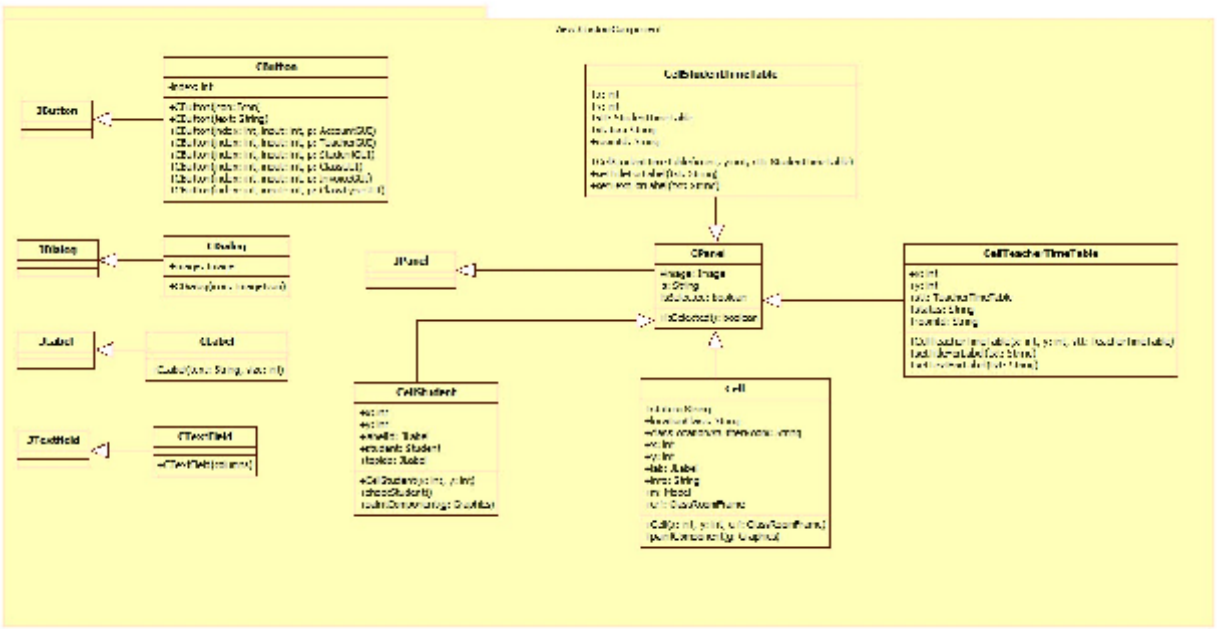
QUICKMANAGE.VIEW.INPUT



QUICKMANAGE.MODEL



QUICKMANAGE VIEW



QUICKMANAGE VIEW.CUSTOMCOMPONENT

4. Design Justifications

- The design is mainly array list data structure. Array list tend to use for accessing elements in an array. The relationship between a model and view follows on MVC model with the model responsible for updating and changing data.

- The design has low coupling between classes and high cohesion between methods in a class. For example, the system has only one model for other controllers to fetch and process data.

4.1 Size and Performance

- Based on the requirement of client, this software is developed as a Java desktop program for a music school.
- The software will be supported to two main kinds of user (Manager and Staff).
- In term of using the software, the user needs to login into the system by username and password which were provided by the Administrator.
- The Manager has the full power to interact with others to do any functions.
- The system also contains the user guide in HTML format for supporting uncommon user.
- The GUI is updated to void the confusion from the user for example

4.2 Quality

- The application has friendly user interface.
- Every feature occurs in the user guide (HTML file).
- Feasible in managing.
- Array List has methods for inserting, deleting etc. So it is useful for this assignment.
- This assignment also includes a lot of consistent regular expression so it help user input the information easily.