2. Overall Description

2.1 Product Perspective

This is a new software product designed for university students to make appointments with instructors. The on campus instructors and the unviersity students are the major two group of users who will benefit from this system. In case both instructors and students may need some extra help, department sectaries are a third class of users to help modify the weekly schedules of instructors and help rearrange schedules when it comes to be the rush periods (near difficult homework deadlines, or exams are coming) and many students want to get appointed but all available periods are occupied already.

2.2 Product Functions

The main events involved in this software includes:

From instructor's side:

post weekly scheduling, modify schedule, modify appointment, delete/cancel student's appointment, and add(rearrange) appointments, and receiving daily emails when there will exist appointments in that specific or coming day.

From Student's side:

add(schedule) appointment, delete(cancel) appointment, receiving automatic emails once add/delete/(be modified by instructor/secretaries) any appointments

From Secetary's side:

modify instructor's schedule, add/delete appointments

2.2.1 Instructor's Functions

2.2.1.1 Post Schedule

2.2.1.2 Modify Schedule

2.2.1.3 Modify Appointments

2.2.1.3 Add Appointments

2.2.1.4 Delete Appointments

2.2.1.5 Receive Email

2.2.2 Student's Functions

2.2.2.1 Add Appointments

2.2.2.2 Delete Appointments

2.2.2.3 Receive Email

2.2.3 Secetery's Functions

2.2.3.1 Modify Schedule

2.2.3.2 Add Appointmenst

2.2.3.3 Delete Appointments

2.3 User Classes and Characteristics

According to main functions from 2.2, there will be three main user groups, and according to group's operation and software interface, each user group function should at least have one sperate Java class definition or class member functions so that the software implementation will include at least three major classes, and each class would need to define all possible/potential member functions which makes all three user groups' operations logical and executable.

Followed are the user classes, member functions and possible/potential arguments:

2.2.1 Instructor's Functions: Class Instructor{}

2.2.1.1 Post Schedule: start date, end date,

2.2.1.2 Modify Schedule: date, schetype,

2.2.1.3 Modify Appointments: date, appflag, applenpre, starttime

2.2.1.3 Add Appointments: date, starttime, applenpre, appflag

2.2.1.4 Delete Appointments: date, starttime

2.2.1.5 Receive Email: date, time, dailyflag

2.2.2 Student's Functions: Class Student{}

2.2.2.1 Add Appointments: date, starttime, applenpre, msg

2.2.2.2 Delete Appointments: date, starttime, applenpre, msg

2.2.2.3 Receive Email: date, starttime, applenpre, msg(reasons)

2.2.3 Secetery's Functions: Class Secretary{}

2.2.3.1 Modify Schedule: date, schetype,

2.2.3.2 Add Appointmenst: date, starttime, applenpre, msg

2.2.3.3 Delete Appointments: date, starttime, applenpre, msg

2.4 Operating Environment

This software will be Window's, Linux and potentially mac cooperate.

The software requirements would be internet explore, or firefox, and internet surfing software needed.

For access to the system, valid university/campus email address and password are required for accessing the software system.

2.5 Design and Implementation Constraints

Right now, we assume/suppose we would get university account system safely accessible for our software. This is only the initial ideas without feasibility consideration and research yet, so we are not sure yet we can completely achieve this goal.

so far no server accessibility consideration yet, we are not sure if we have available resource for this software to be implemented to use yet.

we assume we are going to use university student email account or employee email account as the user check. But we have not consider any potential security issues yet. there may exist some restrictions from the security point of view.

2.6 User Documentation

There are certain rules for all the user groups. and there will be a simple user's guide/manual for each group displayed right behind the scheduling 2-dimensional table, which will least from the most important concerns/issues to least significant ones, so that every user within any group would easily catch the essentials and succcessfully satisfy their requirements.

2.7 Assumptions and Dependencies

Right now, we assume/suppose we would get university account system safely accessible for our software. This is only the initial ideas without feasibility consideration and research yet, so we are not sure yet we can completely achieve this goal.

By designing and reviewing the software specifications so far, we assume the project goal is clear, the implementation language is feasible to achieve the software design goals, and we would be able to fair easily complete the goal.

But from the software application in everyday life, whether the software can be widely used by instructors or students will still depends on the logging systems, software sever availability and user performs. So far, we assume all users are normal, regular user for appointment and studying, solving problem propose, but potential malicous operations may be a threat for the software application.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Right now, we assume/suppose we would get university account system safely accessible for our software. This is only the initial ideas without feasibility consideration and research yet, so we are not sure yet we can completely achieve this goal.

5.2 Safety Requirements

Having not considering if we use student email account, if there will be any safety issues with it.

5.3 Security Requirements

But from the software application in everyday life, whether the software can be widely used by instructors or students will still depends on the logging systems, software sever availability and user performs. So far, we assume all users are normal, regular user for appointment and studying, solving problem propose, but potential malicous operations may be a threat for the software application.

5.4 Software Quality Attributes

Will send of servay to all groups of users to get quality evaluation from users.

5.5 Business Rules

Initiative creation. No copying.