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 the system. In case both instructors and students may occasionally 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:

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

There are mainly three user classes: instructors, students, and department secetary.

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

2.4 Operating Environment

This software will be Window's, Linux and potentially mac operating system compatible.

The only software required would be web browser, like internet explore, firefox etc.

For access to the system, valid university/campus vandalweb id and password are required for logining and accessing the software system.

2.5 Design and Implementation Constraints

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

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

we assume we are going to use university student VandalWeb account as the user authentication check. But we have not implement any potential security issues yet. there may exist some security issues.

2.6 User Documentation

There are quite a few piece of rules for all different user groups. and there will be a simple user's guide/manual for each group displayed right behand the scheduling 2-dimensional table, which will list from the most significant concerns/issues to least relative ones, so that every user within different groups would easily understand the essentials and succcessfully satisfy their requirements.

2.7 Assumptions and Dependencies

Right now, we assume/suppose we would get university VandalWeb account system safely accessible for our software. This is only the initial ideas without deep practical 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 implementation test in everyday life, whether the software can be popularly accepted and widely used by instructors and students or not will still depend on the logining systems, software sever availability and software-user interface and user behaviors as well. We assume all users are normal, regular user for appointment and studying, solving problem propose, having no or very limited controllable bad student users, but potential malicous operations may be a threat for the software application and implementation.

3.4 Communication Interfaces

The communication functions required by this product includes: VandalWeb server, VandalWeb username and password, Web browser.

4. System Features

We will use the Java programming language to create different user classes for the three major user groups, instructors, students and department secartary. In each class, the corresponding class member functions will be declare and defined for manipulation use.

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 System Feature 1: product major classes and definitions

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:

4.1.1 Instructor's Functions: Class Instructor{}

4.1.1.1 Post Schedule: start date, end date,

4.1.1.2 Modify Schedule: date, schetype,

4.1.1.3 Modify Appointments: date, appflag, applenpre, starttime

4.1.1.3 Add Appointments: date, starttime, applenpre, appflag

4.1.1.4 Delete Appointments: date, starttime

4.1.1.5 Receive Email: date, time, dailyflag

4.1.2 Student's Functions: Class Student{}

4.1.2.1 Add Appointments: date, starttime, applenpre, msg

4.1.2.2 Delete Appointments: date, starttime, applenpre, msg

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

4.1.3 Secetery's Functions: Class Secretary{}

4.1.3.1 Modify Schedule: date, schetype,

4.1.3.2 Add Appointmenst: date, starttime, applenpre, msg

4.1.3.3 Delete Appointments: date, starttime, applenpre, msg

4.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use 鈥淭BD鈥?as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

4.2 System Feature 2 (and so on)

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.