COS30041 Creating Secure and Scalable Software

Software Proposal for Distinction (D)

Prepared by: Dennis Siaril (7656203)

Intended Grade: HD

Submission for Feedback (Week 6 – 7) and Final Approval (Week 8, Fri, 5:00pm)

[Note: Based on past experience, it will take at least 2 to 3 submissions to make it right. So, start early]

Instructions - This document is for students aiming to achieve Distinction (D) or above.

For **D grade**, a student needs to do a software application that can demonstrate their skills in developing enterprise application using **a variety of technologies discussed in this subject**. In judging whether your application will be approved or not, the lecturer will be looking into whether the student can utilize various technologies (e.g. as a guide 70%+ of those discussed in the subject) in the application.

For **HD grade**, a student needs to complete the software for D grade as well as a research report. For detailed requirements of the research report, please see the HD_Research_Proposal_Template document in 81_HD_Task_5.4.

Intended Learning Outcomes (extracted from Unit Outline)
1. Build and deploy secure and scalable application using contemporary frameworks
2. Explain and apply strategies, patterns and frameworks to address a range of scalability issues
3. Explain and apply strategies, patterns and frameworks to address a range of security issues
4. Use contemporary tools to evaluate the scalability and security of applications

SOFTWARE PROPOSAL for Distinction

Software Title: Time AI-bler

Introduction
Time AI-bler is a program that uses an AI to make basic decision planning in real-time. In particular, when a user inputs their preferences, the AI will take these preferences into consideration and determine the best output. Furthermore, if there are any changes to the records within the databases, the AI will adjust these outputs in real-time and send out an email detailing every change.

Business Scenario
Time AI-bler is a program that will allow users to create schedules/timetables for their classes. However instead of manually enrolling into class/timeslots, users only need to specify their preferences/availability (e.g. time periods where they are available to attend class). Time AI-bler makes use of an AI to allocate the classes to the best of its ability to match the user's preferences. Furthermore, when a change occurs to a class (i.e. time period changing), the AI will react in real-time and adjust any affected users' timetables accordingly. When the AI makes these decisions, an email is sent out detailing the changes. Though Time AI-bler is specifically used by and for students for timetabling reasons, in truth the program can be expanded upon for task scheduling, budget planning and other resource planning functionality with some adjustments.

Functionalities that you want to implement
Below is a list of functionalities for the software:

F1. Add a student - student will need to provide email and enrol into subjects during creation

F2. Timetable functionality - students should be able to adjust time preferences and request a timetable directly. Otherwise the AI should be able to do this automatically with no preferences and periodically after any changes to the classes occur. The timetable will be displayed basically (e.g. listed on a console), but an email detailing it should be sent out to the specified email as well.

F3. Add subjects and classes - must contain time slots and different types (e.g. lectures, tutorials, labs) for the classes

F4. AI - Create Timetable - when requested, the AI should be able to create a timetable for a student and email it to them, or show them the timetable as well if requested directly.

F5. Edit/delete classes - classes should be able to be modified or removed at any notice. When this occurs, the AI will find any affected students, create a new timetable for each of them, and email them. It will also edit any affected records to reflect the new changes.

Functionality and Technology Traceability Matrix
The following table shows the relevant technologies discussed in this subject that could be used to implement the functionalities as suggested in the Functionalities section above. If you plan to use some other technologies that has not been discussed in the subject, please specify.

At this stage, just the nature of individual software component is enough. There is no need to do "design" on the software component level (e.g. the names and responsibilities of the classes, and how these classes interact with each other). Having said that, it is nice to have but not necessary.

Functionality	Related technologies (discussed in this subject + others, please specify)
	<<e.g. Data Tier – Entity Class using ORM [no JDBC – too old☺]; Business Tier – EJB (Stateless, Stateful, Singleton), JavaMail API; Web Tier – JSF, Managed Beans, AJAX, JSON; Others – Prime-

	Faces [on top of JSF]; use of JSON for Web Services [instead of using DTO for communication between servers]>>
F1	Basic functionality involving ORM. Student's enrolment details should be stored on a relational database (e.g. Student 1 -> Subject 2, Student 1 -> Subject 3, Student 2 -> Subject 1 are all relationships stored as records)
F2	Displaying will require some web technology (e.g. after creation of database, retrieve new records with JSON and display on page). As mentioned, email notifications will also be used, thus the JavaMail API will be used.
F3	Basic functionality involving ORM. Subjects and classes should also have a relational database, as well as students and classes. Classes however can be edited/deleted (perhaps in a separate, simple console program) that will adjust the class records (and the subject-class relation in case of deletion), but not the student-class relation. Instead the AI will adjust the student-class relation when necessary.
F4	Incorporates a basic AI to create the timetable and save the timetable onto the database. The AI will be active in the server, so it should function as a singleton session bean. If users change their preferences and requests a new database
F5	Editing/deleting the classes will be fairly simple, however the AI will have to autonomously let users know of the changes as long as the servers in running, thus introducing a real-time aspect into this enterprise application.