FIT5192 Enterprise and Internet Applications Development Semester 2B, 2019 (Suzhou)

Practical Assignment One Specification

A Java prototype for a custom Web-based product or services recommendation platform

The assignment is worth 25% of your total unit assessment

- This is a pair-based team assignment. Discussion among students on related tutorial questions is encouraged however project teams must work on assignment separately.
- **Please note** that source code and documents will be inspected for similarities between teams and code analysis tools such as MOSS may be used for this purpose.
- Due date: Tuesday 13th August 2019 @ 23:55.
- Late penalty: 10% of the available marks per day.

1. Overview

You have been engaged to design and build a web-based Java prototype for an online products or services platform featuring an extensible recommendation engine.

The overall concept is to develop a web-based recommendation platform where people can buy, loan, hire or procure various products or services. Similar in some respects to well-known cloud based services such as Amazon AWS or Alibaba Cloud (*Aliyun*), this platform can be tailored for the needs of different types of enterprise. In order to evaluate the viability of this system, it is desirable to first build a mock-up of a live system for a particular business case to be determined by your team.

You are required to produce a prototype for the cloud based recommendation platform outlined above with Java enterprise technologies, including JSF, EJB, Persistence API, RESTful Web Services with both web and application clients.

This document includes both specific and more general requirements, thereby providing an opportunity to customise business rules or features appropriate for your particular business concept. If you make any changes then you must **document your assumptions** and ensure that these are reasonable and that the basic technical requirements are still met. Please discuss any proposed changes or additions with your tutor.

The following **Sections 1 to 5** describe the requirements of the complete system, however note that you do NOT have to meet all the requirements to pass the assignment. Please refer to

Section 6 Assessment Guide at the end of the document to find out the different requirements to reach the different assessment levels.

2. Business Entities

This section outlines the main business entities that are required by the proposed system. Depending on your design, you might need more business entities to facilitate the operations of your solution, however the following list has been determined by the project stakeholders to constitute the basic requirements. Note that these are listed under general descriptive headings applicable for a range of business models so suitable entity names should be chosen for your particular solution.

2.1. Platform Users

Three regular types of user will be needed for our cloud-based recommendation platform, each with increasing status and privileges: *Visitors* (the default user type), *Customers* and *Managers*. A fourth *Administrator* type represents a central store or head office, and there can only be one of these (set User ID = 1 for the Administrator). Each type of user shall have the following required information stored within the database:

- User ID number
- Email address
- Password, complying with appropriate validation rules
- Membership level (Visitor, Customer, Manager, Administrator)

For all users other than Visitors, the following personal details must also be recorded:

- Last name
- First name
- Address
- Phone number, complying with a specified numbering system

In addition to the information above, other optional details may be recorded for each user. This may include demographic information such as nationality, age-group, gender, favourite colour or movie. There must be *at least three* optional fields with different data types although the particular choice will depend on the particular business model.

Visitors, only identified by their email, can log in to the system where they can search and view available products and services offered on the recommendation platform. Their search histories may or may not be recorded, depending on your preference.

Visitors cannot buy, loan, hire or procure any products or services however. To be able to perform any of these actions, Visitors must first be upgraded to Customers or Managers. Only Managers or the Administrator can authorise upgrades to Customer membership level, while

only the Administrator can convert a Customer to an Manager. A similar regime also applies for membership level downgrades; see section 3.1 for further details.

Since it is hard to prevent multiple logins by the same individuals, it has been decided that Managers may also acquire products and services on the platform just like regular customers. Moreover, Managers may be offered an incentive bonus in terms of discounts or other service privileges, thereby allowing for organic growth of supported enterprises under a scalable franchising model.

2.2. Products and Services

The cloud based recommendation platform maintains a catalogue of products or services of different types. These can be searched, viewed, bought, hired, borrowed or procured by eligible users according to their membership status. The actual information stored about each type of product or service will vary, however the required attributes in all cases are:

- ID number for each product or service in the catalogue
- A short title
- A thumbnail image
- One or more category labels which can be used for searching the catalogue

Some suggestions for additional information that *may* be included, depending on the nature of the products or services offered and the particular business model employed, are listed below (note that some or all of these may be used as category labels as required for search purposes):

- Price possibly including a discount code
- Availability or number of items in stock
- Loan period and due date
- A detailed description or review
- Color, size and fabrication materials
- Quality ratings or grades
- Author, artist or designer
- Date of manufacture, production or release
- Any other relevant data

As for user entities, there must be *at least three* optional fields with different data types although the particular choice will depend on the particular business model.

2.3. Transactions (e.g. purchases, leases or loans)

At a general level a transaction records the acquisition of one or more products or services by an eligible user, according to the contract terms of the particular business model. The most obvious requirement would be for a customer to pay a total purchase price, although there are countless variations including time-based conditions (such as a library loan) or combination of a

rental fee and a loan period such as would be the case for building equipment hire for example, perhaps allowing for appropriate “add-ons” or variations.

At a minimum, the following information about each transaction shall be recorded in the database:

- ID number of the transaction
- ID number of the responsible user
- ID number(s) of the specific item(s) of the transaction (e.g. products or services)
- The date and time when the transaction took place

Depending on the particular business model, other information relating to the transaction might be needed. For example, a library loan for a number of books would require recording of a return date, a shopping-cart purchase could also record the total price and the prices paid for each item, while an itemised invoice might be needed for professional services delivery.

2.4 Customer satisfaction rating

Finally, to support the development of a user-item preference or recommendation system, it has been decided that an additional entity representing customer satisfaction with practical products or services must be included. During initial planning discussions with project stakeholders, it was assumed that rating could be another attribute for a transaction entity (table column), however in practice transactions made by a user can apply to multiple product or service items (that is a one-to-many relationship) while it is necessary to record a one-to-one mapping between users and items to record customer ratings.

The minimum requirements for this rating entity are

- ID number of the rating
- A numerical customer rating
- Review comments (to be completed at Customer’s discretion)

The simplest approach for the rating system would be to record a 3 valued user ‘like’ (+1) or ‘dislike’ (-1) rating, where a missing entry would be deemed to be equivalent to a neutral rating (0). Alternatively, a five level (Likert) scale might be used.

More complex schemes could even record different categories of user feedback with appropriate survey questions such as “how satisfied were you with the product or service?” This would depend upon the particular business model and needs and is *optional*.

3. System Functionalities

3.1. All Users (including Visitors)

The system permits any user to

- Register as a new user. At a minimum, this requires entering an email address and password according to the following business rules:

    1. Email addresses, required by all users, must be in a valid email format (e.g. <Name>@<EmailProvider>.com).

    2. A password, also required by all users, must conform to the following rules:
        a) must be a minimum of 8 and a maximum of 20 characters long,
        b) contain a mixture of upper and lower case letters,
        c) contain at least one digit (0-9), and must
        d) contain at least one of an arbitrary set of special characters.

    3. New users are assigned a unique ID number in the system. By default, the Membership level of a newly registered user is Visitor.

- Enter other personal details as well as demographic details via an appropriate Web form. The following additional business rules apply to entering personal information on the form:

    4. Last name and first name, if recorded, must not contain any numeric values.

    5. Phone numbers, if recorded, must conform to a valid phone numbering system. For example, for a business based in Melbourne, Australia, the first digit of the phone number must be a 9 and the number must be 8 digits long. On the other hand, if the number is a mobile number, the first digit of the phone number must be 0 and the number must be 10 digits long.

- View and update their own personal information at any time.

  Exception: email addresses and passwords required for registration cannot be updated unless a special process for doing this is included. By default only the Administrator may remove a User account.

- Search for products and services in the catalogue by name, price (if applicable) or by any of a defined set of categorical labels.

- View the details of a catalogued item selected:

    - Search results are firstly displayed in tabular format with heading. Only title, thumbnail, identifier along with business model related information such as availability or purchase price are displayed.

    - The system should also offer the user the option to view the full details from the search results.

Special feature: Product and Services Recommendation Engine

If product and services recommendation logic is implemented then search results will be ordered according to a ranking system as described below:

- If basic preference ranking has been implemented (see **Section 3.5**), searches should simply be ranked according to item popularity among *all* users.

- If similarity based recommendations have been implemented (see **Section 3.6**) then searches should be ranked according to overall popularity of *statistically similar items*.

- If collaborative filtering has been implemented (see **Section 3.7)**, then searches should be further ranked according to predicted preferences *for that user*.

Note that the usefulness of these features will depend to a significant extent on the categorical labels (or possibly ordered, numerical data) associated both with items and users. The choice of these optional fields should therefore be considered carefully.

3.2. Customer

In order for the user to be granted Customer membership status, the following conditions must be met:

1. A validated last name, first name and address has been recorded in the system.

2. A Manager (or the Administrator) approves the registration.

In addition to the functions for all users above, Customers may also

- Place orders and complete transactions according to the rules set out in **Section 2.3**
- Rate purchased or acquired products or services and write review comments about them.
- View their history of transactions.
- View a list of the specific products and services associated with each transaction.

For the purpose of this prototype system, any financial transactions are assumed to have been verified by an external bank.

3.3. Manager

The system permits an Manager to do everything a Customer can do, plus:

- Search for users in the system by the following combination of criteria:

    - ID Number
    - Email
    - Last Name (if known)

  - First Name (if known)
  - Phone number (if known)

- Display search results in tabular format with headings. Only ID number, email, last name, first name and membership status will initially be shown.
- Display the current account details of the user, including information about the current or most recent product or service related transactions if the user has a Customer or higher membership level.
- Upgrade the membership level of a user from Visitor to Customer, or downgrade Customers to Visitors.
- Managers may also update personal details of users (specifically, Name and Address) however they *may not* alter additional demographics or rating related information.

3.4. Administrator

The following special actions are only available to the Administrator (in addition to the functions of regular Managers):

- Add or remove products or services to or from the catalogue.
- Enter and update any attributes or descriptions related to items in the catalogue via an appropriate form. This includes business related attributes such as price, loan period or availability, as applicable.
- Upgrade the membership level of a user from Customer to Manager, or downgrade Managers to Customers.
- Remove a user account from the list.

Note. Downgrading the Membership of a user to Visitor level or removal of a user account may requires one or more prerequisites to have been met, for example:

1. Return of borrowed goods
2. Completion of engaged services
3. Other conditions depending on specific business model

3.5. Basic preference ranking

The user-item rating records allow for a simple form of product and service recommendation. The most obvious way to achieve this would be count up recorded ‘likes’ or ‘dislikes’ and use this to create a ranking table.

Since categorical labels can be used to narrow down a product or services search, this ranking would be applied for the subset of items returned from this search.

Several approaches could be used to implement this feature, such as calculating on demand (for small data sets) or else storing current rankings within an internal data structure or cache.

3.6. Similarity based recommendations

To improve the basic preference ranking system outlined above, the recommendation engine could be updated to determine statistically attractive products and services based upon a combination of user-item ratings **and** similarity between different items.

One advantage of this approach is that products or services similar to ones which were known to be popular however with different categorical labels could be recommended, thereby increasing the variety of offerings.

Such a system could be implemented in a variety of ways, most simply by using the number of different category labels as a similarity measure, The *k* Nearest Neighbour (k-NN) clustering algorithm might also be explored in this context.

Similar to **Section 3.5** above, if this feature was enabled and a user specified search categories, then the resulting ranking would be applied over the subset of items returned from this limited search.

3.7. Collaborative filtering

A further enhancement of the preference matching feature would be tailor the results from the recommendation approaches outlined in either **Section 3.5** or **Section 3.6** in order to consider characteristics of the user conducting the search.

In this case the ranking table would take into account statistical similarities between users. In essence, collaborative filtering works by building a predictive model for what a given group (or cluster) of users are likely to be interested in and then using this information to make suggestions to a new user based on their class membership.

Similar to Section 3.6 above, a simple feature count approach could be used based on differences in terms of demographic category labels, or more sophisticated approaches could be used if this were recorded as ordinal or interval data (such as user age group for example).

Unsupervised learning clustering approaches (such as k-NN) might be used although a relatively simple algorithm could produce very effective results. The example of the *Slope One* algorithm in https://www.baeldung.com/java-collaborative-filtering-recommendations could provide some ideas about how such a feature could be implemented in Java.

4. Design of the Website

The overall web site should have a consistent look and feel. Within each page of the web site there must be a navigation bar that links to other pages.

5. Technology

The prototype described above must be implemented using Java EE technologies (e.g. **EJB**, **JSF**, **RESTful Web Services** and **Persistence API**). A fully functional Java EE system must be developed for the assignment. You must follow the details below to set up the development environment for your solution.

Development Environment:

- JDK 1.7/1.8
- Java EE 7
- Netbeans 8.1 or above
- GlassFish 4.1+
- JavaDB that comes with the Java EE Development Kit

6. Assessment Guide

Appropriate and sufficient data validations and corresponding error messages are to be implemented. The system should also ensure that the user is not allowed to violate the integrity of the existing data in the system.

For every submission, all source code must conform to the Java coding standard and must be well commented. You must also use **@author** tags to identify which parts of the code you wrote.

When you design your solution, you should plan ahead based on the grade you aim at as the design on more advanced level may have an impact on the overall design of your system. ***You must complete all the features and technical requirements on one level before moving up to the next. No marks will be awarded for the higher-grade level's functionality if you have not completed ALL of the functionalities and technical requirements in the successive lower-grade levels.***

** The design of your implementation must use the Object-Oriented programming paradigm and must be flexible and easy to maintain. Ideally your implementation will demonstrate the following good OO design principles:

- Appropriate use of naming conventions and coding standards.
- Little or no noticeable code duplication, "magic numbers" or undeclared literal constants.
- Evidence of method re-use.
- Proper encapsulation and data-hiding.
- Well designed classes and interactions exhibiting *tight cohesion* and *loose coupling*.
- Good use of abstraction, including use abstract classes, interfaces or generics/templates.
- Exception/error handling for critical code sections and "supplier class preconditions".

Pass Level

Features

- Users can register for membership as Visitor by entering email and password in accordance to business rules in **Section 3.1**. Secure login not required at this level.
- Users can update personal details and optional demographics details according to business rules in **Section 3.1**.
- Users can search for products and services by title, type and price.
- Search results are displayed in tabular format with headings. Only title, type and other business specific attributes such as price, number of items or availability are shown.

Technical Requirements

- The system must provide BOTH **web clients** and **application clients** (command-line based or GUI).
- The Web client must be implemented in **JSF**.
- The Application client can be implemented either based on command lines in console window, or using GUI with **Swing** library and it is **preferred** that the GUI be manually created by coding, rather than auto-generated by IDE tools.
- The persistence of the data is managed via **Persistence API**.
- The system must make use of either **application managed entity manager** or **container managed entity manager**.
- Retrieval of data must be done using **JPQL**.

Additional Pass Requirements

1. As this assignment is being undertaken as a team based project this semester, you must complete a **Work Breakdown Agreement** which clearly identifies the responsibilities for each student. Parts of the source code you have written must be identified with **@author** tags matching the task allocations within the WBA. See **Section 8.1 - 8.3** below further details.

2. You are also required to submit a **Learning Summary Report** for the Module as part of the Assignment. A template for the Learning Summary Report is given in the Assignment Section of Moodle. See **Section 8.6** below for further details about what is required for this report.

Credit Level

Features

- All features specified in the **Pass** Level.
- Users are required to login using a username and password to access the system. The password must conform to the validation rules as specified in **Section 3.1** and **3.2**
- After a successful login, users can view the full details of products and services in the catalogue from the search results.

- When searching for products and items on the platform, result are ordered such that highest rated items *within a given search category* are listed first. This corresponds to the **basic preference ranking** approach outlined in **Section 3.5.**
- After a successful login, Managers can search users by ID, last name, first name, phone number or email.
- After a successful login Managers can perform any permitted View/Add/Update/Delete (CRUD operation) on Users, according to business rules and limitations as set out in **Section 3.3**
- Customers can create orders and complete transactions for a single product or service using the platform.

Additional Credit Requirements

As a requirement for Credit level and above, a **Design Report** must be submitted. See **Section 7** for details.

Technical Requirements

- Web client must be implemented in **JSF**.
- ONLY **web client** is required for features required in credit level (**application client** is still required for features required in pass level).
- The persistence of the data is managed via **Persistence API**.
- Make use of either **application managed entity manager** or **container managed entity manager**.
- Use BOTH **Named** and **Dynamic JPQL** for data retrieval.
- The interaction between clients and database must be handled by **EJBs**.

Distinction Level

Features

- All features specified in the **Credit** Level.
- After a successful login, Managers can search users by a *combination* of ID, last name, first name, phone number or email.
- Managers can search Transactions by a *combination* of ID, User and Item related information.
- When searching for products and items on the platform, result are ordered such that statistically similar items with high average ratings *within a given search category* are listed first. This corresponds to the **Similarity based recommendations** approach outlined in **Section 3.6.**
- After a successful login, the Administrator can perform any required View/Add/Update/Delete (CRUD operation) on Users, according to business rules and limitations set out in **Section 3.4**

- After a successful login, the Administrator can perform any required View/Add/Update/Delete (CRUD operation) on Items, according to business rules and limitations set out in **Section 3.4**
- Customers can generate orders and complete transactions for **multiple** products or services using the system.

Technical Requirements

- Web client must be implemented in **JSF**.
- ONLY **web client** is required for features required in distinction level (**application client** is still required for features required in pass level).
- The persistence of the data is managed via **Persistence API**.
- Make use of BOTH **application managed entity manager** and **container managed entity manager**.
- Use BOTH **Named** and **Dynamic JPQL** for data retrieval.
- The interaction between clients and database must be handled by **EJBs**.
- Ability of mapping **inheritance** to database must be demonstrated.
- **Bean validations** must be used to validate data.

High Distinction Level

Features

- All requirements of **Distinction** Level.
- When searching for products and items on the platform, result are ordered such that statistical similarities between items and users are used to recommend products and services *within a given search category*. This corresponds to the **Collaborative filtering** approach outlined in **Section 3.7.**
- When adding items to the system, thumbnail images or other information such as reviews can be obtained automatically via web services (e.g. Google Custom Search, Pixabay or many other openly available web services) instead of being entered manually.
- Securing the Application and encrypting user password using **JAAS API**

Technical Requirements

- Web client must be implemented in **JSF**.
- ONLY **web client** is required for features required in high distinction level (**application client** is still required for features required in pass level).
- The persistence of the data is managed via **Persistence API**.
- Make use of BOTH **application managed entity manager** and **container managed entity manager**.
- Make sure BOTH **Criteria API** and **JPQL** are used for data retrieval.
- The interaction between clients and database must be handled by **EJBs**.
- Ability of mapping **inheritance** to database must be demonstrated.

- **Bean validations** must be used to validate data.
- Consumption of web services must be conducted in **EJBs**.
- Application should be secured using **JAAS API**.

7. Design Report

The Design report should provide a detailed overview of the functionality of your application. This report is an opportunity to demonstrate your team's design decisions and approach which represents a significant part of the assessment.

Your design report must demonstrate originality, good communication skills and present a well thought out application design. It must clearly communicate the abstractions being created, and outline how the functionality is organised.

At a minimum, your design report must include the following elements:

- Overview of your application's goals
- One or more functional/architecture diagrams illustrating how the core aspects of the application fit together.
- Description of core application functionality and how it works.

To facilitate the above requirements, it is recommended that your team includes some or all of the following UML (or comparable design related) diagrams:

1. A **Use Case Diagram** with accompanying descriptions of the separate use cases
2. An **Entity Diagram** with attributes, primary and foreign keys and associations with directionality and cardinality specified
3. A **Class Hierarchy Diagram** detailing any composition or inheritance relationships or interface realizations. Important Java EE container annotations may be included in notes.
4. One or more **Sequence** (or **Collaboration**) **Diagrams**, detailing any more complex interactions in your design.

8. Assignment Deliverables

Your team must submit a zip file consisting of:

1. A **work breakdown agreement** (WBA) indicating the main responsibilities of each team member, in particular which system functionalities or code modules have been developed by each. This may be a simple text file or other short document and must include both student name and the exact **@author** tags as used throughout the source code. To ensure that all students gain practical experience with different Java-EE technologies, ***please check that there is an equitable distribution of coding tasks for each team member across the three main tiers; i.e. web tier (view/controller), application/business tier (model) and database tier***. It is also advisable for team

members to verify their partner's work in order to understand it sufficiently well that they are able to explain it *in detail* during a code walk-through / interview.

2. A draft WBA must be prepared by **Tuesday 6th August 2019** and discussed with your tutor during the labs. The WBA will be referred to during progress of the assignment and for assessment particularly if there are questions about contributions by team members.
3. Source codes; please ensure that **@author** tags (exactly matching those included for each student within the WBA document) are included *throughout* the source code.
4. A script for generating your database, tables and testing dataset (for recreating the DB) with the following naming convention: Database name: **idTeamName**. For instance, if your team name is "**JavaTerminator**", then your database should be named **idJavaTerminator**. Please set your DB access account and password as both fit5192.
5. Instructions (the README file) for running the application (including the name, username and password of the database used);
6. Your **Learning Summary Report in PDF format**. (Note: you may **either** submit one learning summary report per team member, or else create one **combined** report for your team in which individual contributions and personal comments are clearly indicated.)
7. In addition to the above, a **Design Report in PDF format**, if you are aiming for a Credit or above. (**Note:** a draft version may also be submitted separately prior to the due date.)

All the files above should be uploaded to Moodle as a zip file and use the following naming convention: FIT5192A1_TeamName.zip. For instance, if your team name is "0**Error0Warning**" then the file you submit should be named **FIT5192A1_0Error0Warning.zip**.

You must submit your work by the submission deadline on the due date (a late penalty of 10% per day of the possible marks will apply - up to a maximum of 100%). Students will be **interviewed** during their normal tutorial classes right after the assignment due date. You can expect to be asked to explain your original design, discuss design decisions and alternatives, and possibly to modify your code as required. A list of interview times will be posted on the FIT5192 Moodle forum and/or circulated in class.

9. Plagiarism

Before submitting your assignment, please make sure that you have not breached the University's plagiarism and collusion policy. It is the student's responsibility to familiarize themselves with the contents of these documents.

Plagiarism and collusion are very serious offences. In cases where cheating has been confirmed, students would be severely penalized, from losing all marks for an assignment, failing a unit or facing disciplinary action at the Faculty level. While we would wish that all our students adhere to sound ethical conduct and honesty, please ensure that you are well acquainted with the following policy from the Plagiarism Procedures of Monash, available at

http://www.policy.monash.edu/policy-bank/academic/education/conduct/student-academic-integrity-managing-plagiarism-collusion-procedures.html