



## **FIT5042 Design Report (Major Tasks)**

### **Banking Enterprise Application**

Student Name: Duo Pan

Student ID: 27554074

Tutor Name: ABM Russel

Tutorial Time: Mon 18:00-20:00

Lecturer: Chris Messom

## Table of Contents

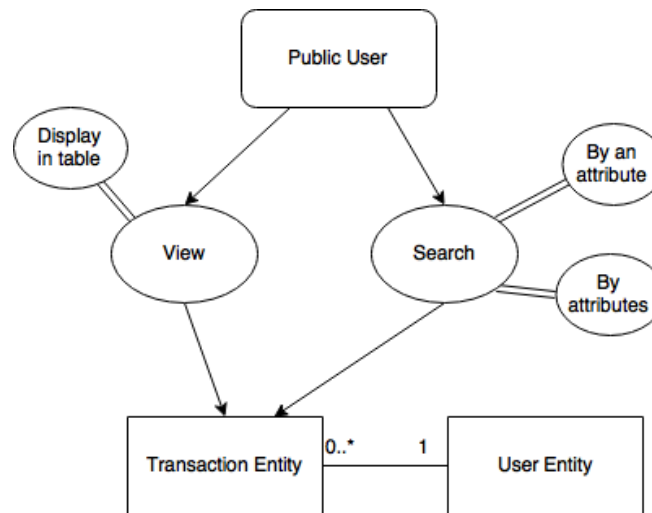
<b>1. Overview .....</b>	<b>3</b>
<b>2. Functional diagram .....</b>	<b>3</b>
<b>3. Core program functionality .....</b>	<b>4</b>
<b>4. Usability Design Review .....</b>	<b>7</b>
<b>5. Checklist of site functionality.....</b>	<b>8</b>
<b>6. User stories .....</b>	<b>8</b>
<b>7. Entity relation diagram.....</b>	<b>8</b>
<b>8. Data dictionary .....</b>	<b>8</b>

## 1. Overview

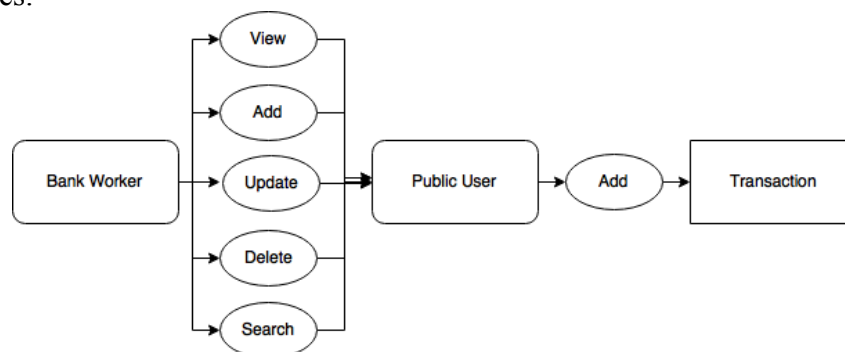
This assignment is to build a web-based Java prototype of a Banking Enterprise Application as well as a desktop client application version. Some Java enterprise technologies should be used in the assignment: JSF, RESTful web services, EJB, Persistence API and so on.

The basic function is that users can login in the system by their username and password. System will then redirect the user to the specific page. If the user is a public customer, he/she can view and edit his/her own information, and can view, create, sort, multiple search their transaction records. If it is a bank worker, he/she can view, create, update, delete and search users.

## 2. Functional diagram

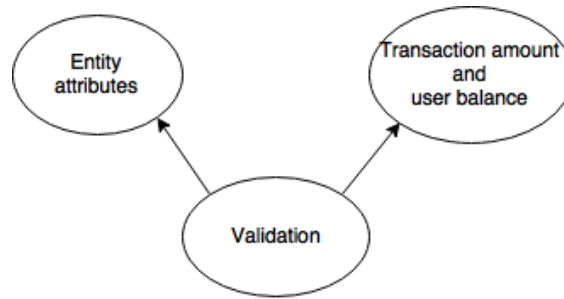


- There are two entities in database, user and transaction. One user has 0 to many transactions and one transaction can only belong to one user.
- Public user can view his/her transactions, the results are displayed in table.
- Public user can search his/her transactions, either by one attribute or multiple attributes.



- Bank worker can view/add/update/delete users, and can search by multiple attributes of a user.

- Publish user can create new transaction, such as deposit or withdraw money. So I add balance attribute to user entity, and add amount attribute to transaction entity.



- When adding records to database, the system will do validation on each attribute as the assignment required. Such as the format of email.
- When a user withdraw money, the amount cannot exceed his/her balance.

### 3. Core program functionality

Use BootsFaces library to build the page, which is designed for JSF and based on Bootstrap. It provides a table control, and through changing its attributes, I can get functions I need.

- View all data in table, click view button, can see detail of this transaction record.

My transaction Records

Search:  [+ Create](#) [My Information](#)

No	Name	Type	Description	Amount	
1	a	Deposit	ff	10.0	<a href="#">View</a>
2	fd	Deposit	da	10.0	<a href="#">View</a>
3	ww	Deposit	fdaf	10.0	<a href="#">View</a>
4	fdafd	Deposit	11	10.0	<a href="#">View</a>
6	af	Deposit	afd	10.0	<a href="#">View</a>

No  Name  Type  Description  Amount

Showing 1 to 5 of 5 entries

[Column visibility](#) [Excel](#) [PDF](#) [Print](#)

[Previous](#) [1](#) [Next](#)

**Transaction Details**

No.

Name

Type

Description

Amount

[← back](#)

- Can choose which attributes to show on table.

**My transaction Records**

Search:  [+ Create](#) [My Information](#)

Name	Amount	
a	10.0	<a href="#">View</a>
fd	10.0	<a href="#">View</a>
ww	10.0	<a href="#">View</a>
fdaf	10.0	<a href="#">View</a>
	10.0	<a href="#">View</a>

Column visibility Excel PDF Print

Previous 1 Next

- Multiple search. I put a search textbox under each column, so users can input data in the corresponding search textbox to do multiple search. The example search attributes by both name and amount.

**My transaction Records**

Search:  [+ Create](#) [My Information](#)

No	Name	Type	Description	Amount	
1	a	Deposit	ff	10.0	<a href="#">View</a>
4	fdaf	Deposit	11	10.0	<a href="#">View</a>
6	af	Deposit	afd	10.0	<a href="#">View</a>

No  a  Type  Description  10

Showing 1 to 3 of 3 entries (filtered from 5 total entries)

Column visibility Excel PDF Print

Previous 1 Next

- Navigation: Provide buttons for user to view transaction details and come back to main page.
- Bank worker CRUD user functions:  
View all users / delete a user / view one user's all transactions:

The screenshot shows a web browser window with the URL 'localhost'. The browser tabs include 'fit5042D.xml - draw.io', 'DuoPan/DrawIO', 'FIT5032-S2-2017: Draft C Code Task Details (...)', and 'Bank System'. The page title is 'All User Records'. Below the title is a search bar and two buttons: '+ Create' and 'My Information'. The main content is a table with columns: ID, Last Name, First Name, Email, Password, Type, Address, Phone, Balance, and three action buttons: View/Edit, Delete, and Transactions. The table contains two entries. Below the table are search filters for each column and a pagination bar showing 'Showing 1 to 2 of 2 entries' with buttons for 'Column visibility', 'Excel', 'PDF', 'Print', 'Previous', '1', and 'Next'.

ID	Last Name	First Name	Email	Password	Type	Address	Phone	Balance			
1	pan	duo	pd@qq.com	123	Public	aaa	98888888	1050.0	<a href="#">View/Edit</a>	<a href="#">Delete</a>	<a href="#">Transactions</a>
2	d	a	a@qq.com	1234	Bank Worker	aa	98877665	1000.0	<a href="#">View/Edit</a>	<a href="#">Delete</a>	<a href="#">Transactions</a>

Showing 1 to 2 of 2 entries

Column visibility Excel PDF Print

Previous 1 Next

### Create/Edit a user:

The screenshot shows a web browser window with the URL 'localhost'. The browser tabs include 'fit5042D.xml - draw.io', 'DuoPan/DrawIO', 'FIT5032-S2-2017: Draft C Code Task Details (...)', and 'Create New User'. The page title is 'Create New User'. The form contains the following fields: User ID, Last Name, First Name, Email, Password, Type (dropdown menu), Phone Number, and Address. At the bottom are 'cancel' and 'save' buttons.

User ID \*

Please enter User ID

Last Name \*

Please enter last name

First Name \*

Please enter first name

Email \*

Please enter email

Password \*

Please enter password

Type \*

Public

Phone Number \*

Please enter phone number

Address \*

Please enter address

cancel save

- Public user can create new transaction

The screenshot shows a web browser window with the URL 'localhost'. The browser tabs include 'fit5042D.xml - draw.io', 'DuoPan/DrawIO', 'FIT5032-S2-2017: Draft C Code Task Details (...)', and 'Add'. The page title is 'Add Transaction'. The form contains the following fields: No., Name, Type (dropdown menu), Description, Amount, and Balance. At the bottom are 'cancel' and 'save' buttons.

No.

Please enter Transaction No.

Name

Please enter Transaction Name

Type

Deposit

Description

Please enter Transaction Description

Amount

\$0.00

Balance

\$1,050.00

cancel save

- Validation: when user click save button, the system will do validation and prompt errors on the screen in red text.

### Add Transaction

No.	88
Name	good
Type	Withdraw
Description	test
Amount	\$9,999,999.00
Balance	\$1,050.00

[← cancel](#) [✓ save](#)

• Balance is not enough.

## 4. Usability Design Review

- Navigation: Each page has good navigation buttons to the other pages, they are colorful and easy to find;
- Familiarity: This web site did not adopt any strange layout, so it is familiar with most people;
- Consistency: This web application can run at different browsers.
- Error Prevention: User inputs are check before store into the database;
- Feedback: When mouse moves over table rows, it will highlight.
- Visual Clarity: Data table makes the web site easy to read and use. Red color on delete button also remains user be careful when clicking. The whole page is clear and tidy.
- Flexibility: Many buttons in a page, so that user can easily go to a page they want. All the pages are connected in graph structure instead of linear.

## 5. Checklist of site functionality

<b>1. Credit Functionality</b>	✓
Search for Transaction by	✓
Transaction Name,	✓
Transaction No	✓
Transaction Type	✓
Results with tabular format with heading.	✓
Option to view the full details	✓
<b>2. Distinction Functionality</b>	✓
Bank Workers can: View	✓
Add	✓
Update	✓
Delete Users	✓
Members of Public can make Transactions	✓
<b>3. Technical Requirements</b>	✓
<b>Credit</b>	✓
JSF web clients	✓
Persistence API	✓
Application managed entity manager or container managed entity manager.	✓
<b>Distinction</b>	✓
Web client is required	✓
Interaction between clients and database handled by EJBs	✓
BOTH Criteria API and JPQL	✓
<b>Audit</b>	✓
No breaking of copyright	✓

## 6. User stories

I am not sure about this part. Need discussion on tutorial. Thank you.

## 7. Entity relation diagram

## 8. Data dictionary