The Banking Project

Mara Munoz

CST-452 Capstone Project Final

Grand Canyon University

Instructor: Professor Michael Landreth

Date: 05/28/23

**ABSTRACT**

This project will handle transactions such as deposit and withdrawal for different client’s bank accounts. This will be written in Java and will contain connection to a MySQL database using JDBC. There will be login functionality, and the user can reach either the login success or failure message. Next the user will have three options for deposit, or withdrawal.

Deposit - If the user selects this option, they can enter how much to deposit. When they enter, that amount is then added and saved to their account in the database.

Withdrawal - If the user selects this option, they can enter their withdrawal amount. When they enter, the backend code will test this amount against the users account balance and if it overdrafts the user it will not go through and will either show an error message or navigate to an error page.

Lastly, there will be a logout option which returns to the login menu.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 05/05/23 | Mara Munoz | Initial draft for review/discussion |
|  |  |  |
|  |  |  |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes No

**TABLE OF CONTENTS**

**sPRINT PLANNING………………………………………………………………………………………………………………………………4**

**Design Overview 4**

**SOURCE CODE LISTING ……………………………………………………………………………………………………………………….5**

**Detailed High-Level Solution Design 5**

**Detailed Technical Design 6**

**Appendix A – Technical Issue and Risk Log 7**

**Appendix B – References 8**

**Appendix C – External Resources 9**

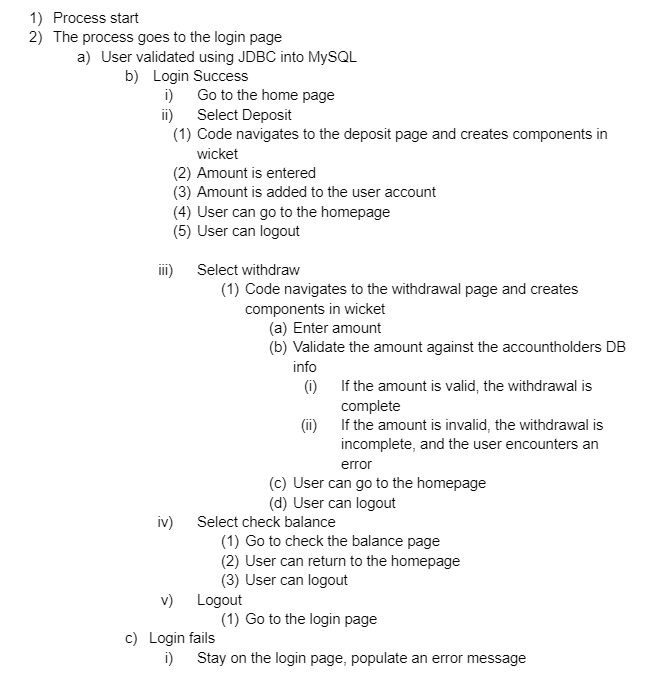
**Sprint Planning**

A screenshot of a computer

Description automatically generated

**Design Introduction**

Code process



A picture containing text, diagram, screenshot, line

Description automatically generated

Flow chart:

A diagram of a website

Description automatically generated with low confidence

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Deliverable Acceptance Log | | | | | |
| ID | Deliverable Description | Comments | Evaluator (internal or external as applicable) | Status | Date of Decision |
| 1 | JDBC driver | For MySQL connection |  | Complete | 05/04/23 |
| 2 | Wicket Requirement | For front end |  | Complete | 05/04/23 |
| 3 | Java Requirement | For logic |  | Complete | 05/04/23 |
| 4 | MySQL Requirement | For data needs |  | Complete | 05/04/23 |
| 5 | JDBC service classes | For using JBDC to get/store data |  | Complete | 05/04/23 |
| 6 | SLF4J Logging | Logging what is currently happening in the application |  | Incomplete | 05/04/23 |

**SOURCE CODE LISTING**

* main.java
  + com.mycompany
    - HomePage.java
    - HomePage.html
    - WicketApplication.java
  + service
    - JDBCConnection.java
    - Service.java
  + scripts
    - Script.sql
* Test.java.com.mycompany
  + Start.java
  + TestHomePage.java
  + TransactionTest.java

**Detailed High-Level Solution Design**

A picture containing text, screenshot, font, diagram

Description automatically generated

**A picture containing diagram, screenshot, text, line

Description automatically generated**

|  |  |  |
| --- | --- | --- |
| Proof of Concepts | |  |
| **Description** | **Rationale** | **Results** |
| 1. The new program will have a wicket user interface | Wicket is stable to support high usage | Excellent interface |
| 2 - The new program will allow users to log out and login at any time | Users need to know their information is secure | Customer security |
| 3 - Database values saved of transaction value, and before and after transaction balances | Users can see all calculations on the screen | Customer satisfaction in the details |
| 5 - |  |  |

|  |
| --- |
| Hardware and Software Technologies |
| 1 - MySQL |
| 2 - Wicket |
| 3 - Java |
| 4 - JDBC |
| 5 - SLF4J Logging |

**Logical Solution Design:**

- Completed:

- Project overview

- High-level solution design

- Project success measures and scope

- Technical Requirements

- Functional requirements

- User interface design

- Nonfunctional requirements

- Logical system design

- Detailed solution design

- Pseudo code

- Design overview

- Technical requirements

- Code

- Test plan

- Test cases

Week 5:

* Adding more functionality to the code
* Test cases
* Fine tuning the DB
* Added user stories NFR and FR
* Implementing bootstrap in the UI
* Updating diagrams
* Source code listing

Week 6 (this week):

* Adding SLF4j logging
* Fine tuning the UI
  + Centering
  + Button style
  + Background color
* Complete coding and testing
* Improve everything in the previous week

Week 7-8:

* Prepare the project for presentation
* Add a readme file
* Demonstration of project

**Physical Solution Design:**

- System architecture will contain both back and front end in the Java package of code. The front-end wicket framework lives right next to the classes.

- The data model will contain two models—user and account. See at the end of this file for the diagram documenting them.

- The user interface design consists of 4 panels: Login, homepage, deposit, and withdraw. See the end of the file for the wireframes.

- Database will be held in MySQL and connected to Java using JDBC driver.

**Detailed Technical Design**

**General Technical Approach:**

After several brainstorming sessions we determined the best route to take for languages. We will be going with Java as the main language and will use the Wicket framework for the front end. MySQL will be used for the database, and JDBC. These are all free options, which makes it very appealing from a financial standpoint to our stakeholders. The project will be split up into three main directories: models, view, and controller in order to separate the layers.

**Key Technical Design Decisions:**

Java was picked due to its established presence and security in many software. JDBC was selected for its place between the models in the application and MySQL as well as being selected for its simplicity. MySQL was selected because it is a reliable relational database that is free of cost compared to SQL. Wicket was selected due to previous experience and its involvement in live banking software.

**Database ER Diagram:**

A picture containing diagram, screenshot, text, line

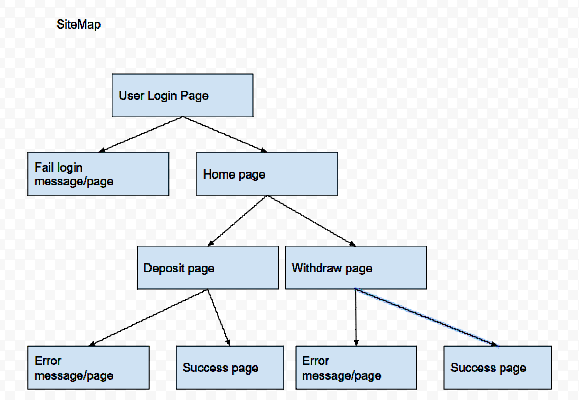
Description automatically generated

**Database DDL Scripts:**

A screenshot of a computer program

Description automatically generated with medium confidence

**Sitemap Diagram:**



**User Interface Diagrams:**

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

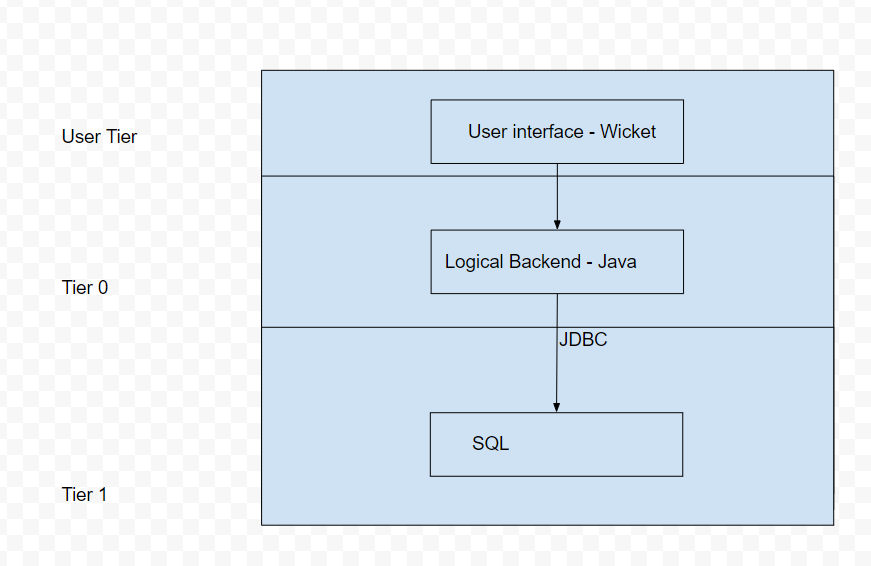
**UML Diagrams:**

**Service API Design:**

- JDBC provides abstraction as an API which allows us to connect to MySQL

- MySQL uses a driver downloaded from oracle

- The framework Wicket is used by adding to the pom.xml



**NFR’s (Security Design, etc.):**

|  |  |  |
| --- | --- | --- |
| **Use Case or User Story** | **Approval Date** | **Justification** |
| NFR1- User login validation | 5/6/23 | User login validated against database information |
| NFR2- Create navigation between pages | 5/6/23 | User must be able to navigate |
| NFR3- Validate user withdrawal amount | 5/6/23 | In order to stop an overdraft, amounts must be validated |
| NFR4- Success in drivers, frameworks and plugins | 5/6/23 | Needed for success of project |

**Operational Support Design:**

Logging will be throughout the entire project. It will log the class where the message is originating from and it will produce a message for either info or error. If in runtime there are caught exceptions, there will be logs displaying the error, and potentially an additional string for better understanding.

**Appendix A – Technical Issue and Risk Log**

* Use the template to identify and monitor project issues and risks.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues and Risk Log | | | | | | | | |
| **Issue or Risk** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| I/R | What is the issue or risk? | How will this impact scope, schedule, and cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

**Appendix B – References**

*List all Project Documentation References*

*List all references using APA style*

**Appendix C – External Resources**

|  |  |
| --- | --- |
| **GIT URL:** | [maukamnm/seniorBankingRepo (github.com)](https://github.com/maukamnm/seniorBankingRepo) |
| **Hosting URL:** | *The Hosting URL (if applicable).* |