Graphical user interface, text, application

Description automatically generated

[**SOEN 6431**](https://github.com/manimayan/SOEN_6431_Deja_Vu) **- Software Comprehension and Maintenance**

**Summer 2022**

**Deliverable 1- Reengineering Opportunity**

[GitHub link](https://github.com/manimayan/SOEN_6431_Deja_Vu.git)

***Authors***

Manimaran Palani

Iphigenia Pappas

Heet Patel

Kevinkumar Patel

Venis Patel

https://www.overleaf.com/project/6285ba87cc2d8b26da21563d

Table of Contents

[**1.** **Abstract** 3](#_Toc105412388)

[**2.** **Introduction** 3](#_Toc105412389)

[**3.** **Project Deliberation** 3](#_Toc105412390)

[3.1. Communal Work 3](#_Toc105412391)

[3.2. Responsibility Matrix 3](#_Toc105412392)

[3.3. Candidate System Selection Criteria 3](#_Toc105412393)

[**4.** **System Descriptions** 4](#_Toc105412394)

[4.1. **Online Banking System** 4](#_Toc105412395)

[4.2. **Currency Converter** 5](#_Toc105412396)

[4.3. **Hotel Management System** 6](#_Toc105412397)

[4.4. **Product Management System** 7](#_Toc105412398)

[4.5. **E-Book Stall** 8](#_Toc105412399)

[**5.** **Candidate System Description** 9](#_Toc105412400)

[**6.** **Candidate System Rationale** 9](#_Toc105412401)

[**7.** **References** 9](#_Toc105412402)

# **Abstract**

For a software system to be able to successfully sustain in the market, constant evolution becomes of utmost importance. Evolution refers to continuous developments and updates in the originally created software system to match the ever-changing market demands at any given point of time. This process of refining the software to keep it abreast of the trends and demands is termed as Software Maintenance. To carry out a successful maintenance by implementing positive changes in the software systems, the analysis of source code and documentation and having a comprehensive understanding of the system holds the most importance. For this assignment, we choose a candidate R from five different repositories chosen by respective team members. We dive into the detailed analysis of the ‘undesirables’ in the repositories using TeamScale before tapering down to the candidate R. We further explain the repudiation rationale for the same. The chosen candidate R provides us with the opportunity to reengineer the software program by rectifying the ‘undesirables’ and thus improving its efficiency and maintainability.

# **Introduction**

Software Maintenance plays a massive role in always keeping the software in a ‘good shape’. This in turn benefits the maintainers of the software in by-passing the whole cost and hassle of creating  
a new software system from scratch after being outdated and aids in keeping it up to date  
according to the market needs. Maintenance also ensures solid quality of code and maintainability.  
This is very useful in safeguarding the software system from having potential ‘undesirables’ which  
can further lead to a decline in quality and efficiency of the source code.  
A reengineering process on the source code and documentation of a software can further widen  
the horizon for itself in terms of its usability, maintainability, and efficiency. We carry out an initial  
analysis using TeamScale of five different repositories by taking note of five unique ‘undesirables’  
for each one of them. We then analyse, based on the quality of the source code and documentation,  
the reasons for a repository to be an ideal candidate R for reengineering.  
Upon successful identification of the candidate R, we buckle down to investigating the ‘unde-  
sirables’ in more detail. We further plan on materializing the solutions to the problems observed  
in the candidate R by carrying out reengineering operationalization in the upcoming assignment.

# **Project Deliberation**

## Communal Work

## Responsibility Matrix

## Candidate System Selection Criteria

***Team’s Programming capability***

We assessed the experience that all the team individuals have and chosen that Java would be the foremost ideal programming language as all the team members have similar experience using Java as the development language. Any application that doesn’t utilize Java given lesser priority.

***Complex Architecture***

An application that's less complicated to setup is mostly preferred because it will reduce the hours to spend on setting up the project rather spending it on analysing the project to meet the metrics of candidate system.

***Code quality -*** The lower the code quality, the more imperative it is to make improvements in it. Typically, its determined by each team member, who rates the project’s quality in the scale of identifying 25 unique undesirables. Each group member has assessed the code using the tool TeamScale and agreed collaboratively upon choosing the candidate system.

# **System Descriptions**

## **Online Banking System** Manimaran Palani - 40167543

**Project Description**

The Online Banking System is a banking portal on the web which manages the customer profile and manages their transactions. It is highly scalable and secured with the help of Spring Security. The main feature of this project is validation of login form, viewing customer proﬁle, viewing transaction details of the customer ,Viewing balance of the customer, Approval of the changes in the address by the customer. The core objective of this project is to maintain a personal account in the bank. The system also provides the access to the customer to create an account, deposit/withdraw the cash from the account, also to view reports of all the accounts available.

**System Stack** : Spring Boot, Spring Security, Thyme leaf, Spring Data JPA, Spring Data REST, JavaScript, jQuery.

**System Source Code** : <https://github.com/ryhan000/Online-Banking-System.git>

**Repudiation Rationale**

The above project is chosen as the candidate R system due to various factors and the details are provide in the section (6). The below picture depicts the software quality report generated by integrating the application with the tool TeamScale which analyse, monitor, and improve the quality of the application.

Graphical user interface, application

Description automatically generated

Figure 1: Code Quality Report of Online Banking System - TeamScale Dashboard

## **Currency Converter** Iphigenia Pappas - 40077089

**Project Description**

Currency converter (or currency exchange) is a mini project coded in Java programming language. This simple application provides a web-based interface for exchanging/converting money from one currency (say $) to another currency (say €). It is simply a calculator-like app developed using Ajax, Java servlets web features. In this application, there is regular update about currency of every country by which it displays present currency market value and conversion rate. Such application can be used by any user, but it is mainly useful for business, shares, and finance related areas where money transfer and currency exchange take place daily.

**System Stack :** Ajax, Java servlets

**System Source Code :** https://github.com/projectworldsofficial/currency-converter-in-java.git

**Repudiation Rationale**

This repository had many issues, however the large majority of things that stuck out were features that were not implemented properly. I noticed many defects. Firstly, on top of all the maintenance that needs to be done to the code, the bigger issue was a performance issue, this task could have been done in O(1) rather than O(n) by using a switch statement. The interface defects also bothered me, as much work could have been done to make it more user friendly and allow it to perform multiple conversions at once. There are also multithreading defects as we cannot execute multiple tasks at once. The problem with multithreading, however, is that sometimes there is the condition of deadlock where starvation is created, and if not handled and debugged properly, can end up leading to a failure of the system. Most of the issues had to do with the design of the software rather than the actual code itself, and as this is not the end goal of this assignment (to add enhancements and change the software) therefore we rejected this Software chosen.

Graphical user interface

Description automatically generated with medium confidence

Figure 2 : Code Quality Report of Currency Converter - TeamScale Dashboard

## **Hotel Management System** Heet Patel – 40179213

**Project Description**

Hotel Management System is developed in java platform. This is a Web Base Management System developed for Managing Hotel Day. This application covers all activities in the hotel including Registrations, Employee Management, Bill Printing, Account Control, Result analysing, Food and Beverage Details, and Issuing Rooms. The system role is divided into two types: manager (administrator) and employee (ordinary user), of which manager (administrator) has the right to view all bookings, delete rooms, view employees, add employees and other functional permissions Employees (ordinary users) have the right to view available rooms, customer reservations, modify reservations, delete reservations, register new customers and other functional permissions (tasks are self-drafted).

**System Stack :** HTML5, CSS3, JavaScript, Bootstrap3, JDK1.8 + Spring Boot, Mybatis (Persistence Layer Framework), Druid (Database Connection Pool)

**System Source Cod**e : <https://github.com/jsanhuo/HotelManagementSystem.git>

**Repudiation Rationale**

**Graphical user interface

Description automatically generated**

Figure 3 : Code Quality Report of Hotel Management System - TeamScale Dashboard

## **Product Management System** KevinKumar Patel - 40194915

**Project Description**

It is a marketplace where customer can place order and Admin can manage inventory of products, view order and generate the report.

**System Stack :** Java 8, Spring MVC, Spring Data, Spring Boot, Hibernate JPA, H2 In-memory Database, Maven for Dependency Management.

**System Source Code :** <https://github.com/anantjain6/ProductManagementSystem.git>

**Repudiation Rationale**

Graphical user interface, chart

Description automatically generated with medium confidence

Figure 4 : Code Quality Report of Product Management System - TeamScale Dashboard

## **E-Book Stall** Venis Patel - 40170617

**Project Description**

The application is built to create an online platform to sell books, The application comprises of functionalities like maintaining books selling history, adding, and managing books, updating the availability of books, filtering the books based on author or the publisher and the simulation of online payment method to buy the book. The application is built in Generic Servlets in Java, JDBC and MySQL.

**System Stack :** Html, CSS, JavaScript, Java [JDK 8+], JDBC, Servlet, MySQL, Apache Maven.

**System Source Code :** <https://github.com/shashirajraja/onlinebookstore.git>

**Repudiation Rationale**

**Chart

Description automatically generated**

Figure 5 : Code Quality Report of E-Book Stall - TeamScale Dashboard

# **Candidate System Description**

The Online Banking System is a banking portal on the web which manages the customer profile and manages their transactions. It is highly scalable and secured with the help of Spring Security. The main feature of this project is validation of login form, viewing customer proﬁle, viewing transaction details of the customer ,Viewing balance of the customer, Approval of the changes in the address by the customer. The core objective of this project is to maintain a personal account in the bank. The system also provides the access to the customer to create an account, deposit/withdraw the cash from the account, also to view reports of all the accounts available.

# **Candidate System Rationale**

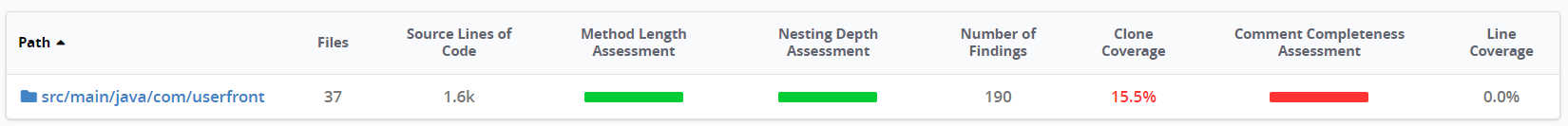


Figure 6 : overall assessment generated for Candidate R online banking system from TeamScale

Graphical user interface, application, table

Description automatically generated

Figure 7 : Category of Findings generated for Candidate R online banking system from TeamScale

Graphical user interface, application, email

Description automatically generated

Figure 8 : Quality Metrics used to assess Candidate R online banking system from TeamScale

Graphical user interface, application

Description automatically generated

Figure 9 : Code Quality Report of Online Banking System - TeamScale Dashboard

# **References**

[1] "Teamscale Documentation" [Online]. Available: https://docs.teamscale.com/#why-teamscale-is-different.

[2] D. Korolev, "How do you define code quality?" 09 02 2014. [Online]. Available: https://www.quora.com/How-do-you-define-code-quality.

[3] "ISO/IEC 25010:2011," 03 2011. [Online]. Available: http://www.iso.org/iso/catalogue\_detail.htm?csnumber=35733.

[4] "ISO/IEC 9126-1:2001," 06 2001. [Online]. Available:

http://www.iso.org/iso/catalogue\_detail.htm?csnumber=22749.

[5] R. &. Norvig, The intelligent agent paradigm, 2003, pp. 27, 32–58, 968–972 .

[6] "XML Tutorial," 2017. [Online]. Available: <http://www.w3schools.com/xml/>

[7] "What is the optimal size of a software development team," 16 05 2009. [Online].

Available: http://stackoverflow.com/questions/872103/what-is-the-optimal-size-of-a-software-development-team.

