NEU-DPSA-2018-2019-LAB-REPORT-08-1

Group member:

20165164 Nora Yang

20165187 Emily Chen

20165243 Rown

Contents

[1. What we’ve done 2](#_Toc533634824)

[2. Design 3](#_Toc533634825)

[2.1 Project Structure 3](#_Toc533634826)

[2.2 Description 3](#_Toc533634827)

[2.3 Class Diagram 4](#_Toc533634828)

[2.4 Sequence Diagram 4](#_Toc533634829)

[3. Unit Testing 5](#_Toc533634830)

[4. Deployment 6](#_Toc533634831)

[5. Screenshots of GUI 7](#_Toc533634832)

[5.1 Calculate mortgage 7](#_Toc533634833)

[5.2 Make a payment 7](#_Toc533634834)

# 1. What we’ve done

- 2 patterns used: Observer & Strategy

- Class diagram and sequence diagram (in report and as .jpg files)

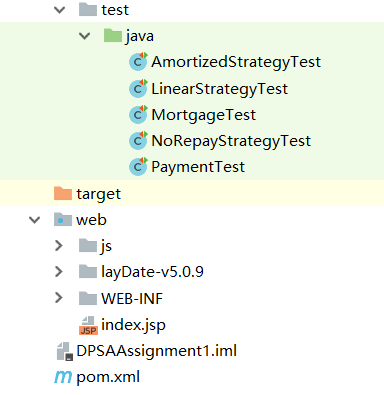
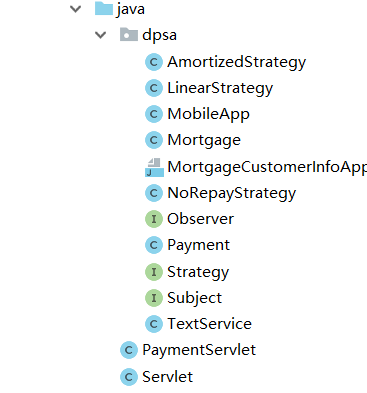
- JUnit testing

- GUI design using JSP

- Build using Maven, pom.xml and the .war file are provided

# 2. Design

## 2.1 Project Structure



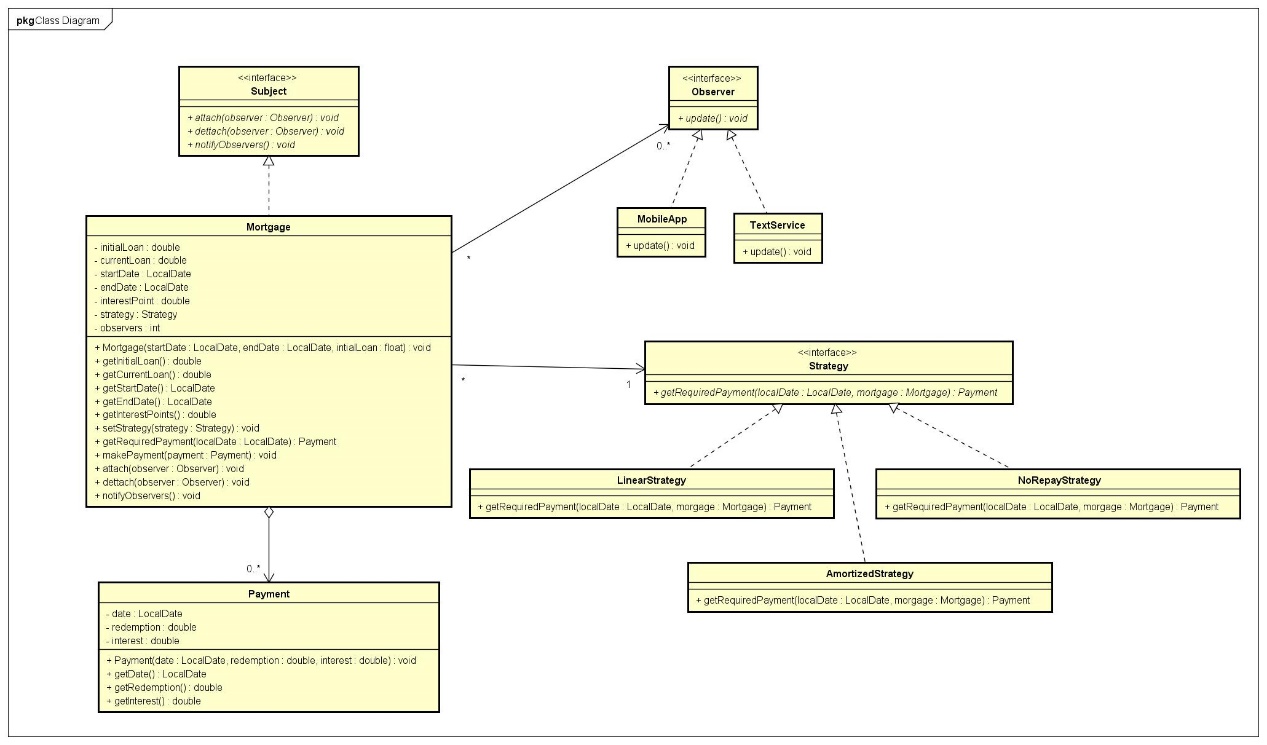
## 2.2 Description

In our design, Strategy Pattern is used to calculate payments. Mortgage has a Strategy. Given a queryDate, a Mortgage object will ask its concrete Strategy object to calculate the required payment according to its algorithm and return a Payment object.

And the Observer Pattern is used to notify related services when payments are made. Mortgage, the concrete Subject, has a List<Observer>. Every time a user makes a payment, mortgage.makePayment() is called and the Mortgage object will notify its observers by iterating the list and calling observer.update(). We use dummy objects (of class MobileApp and TextService) as concrete Observers.

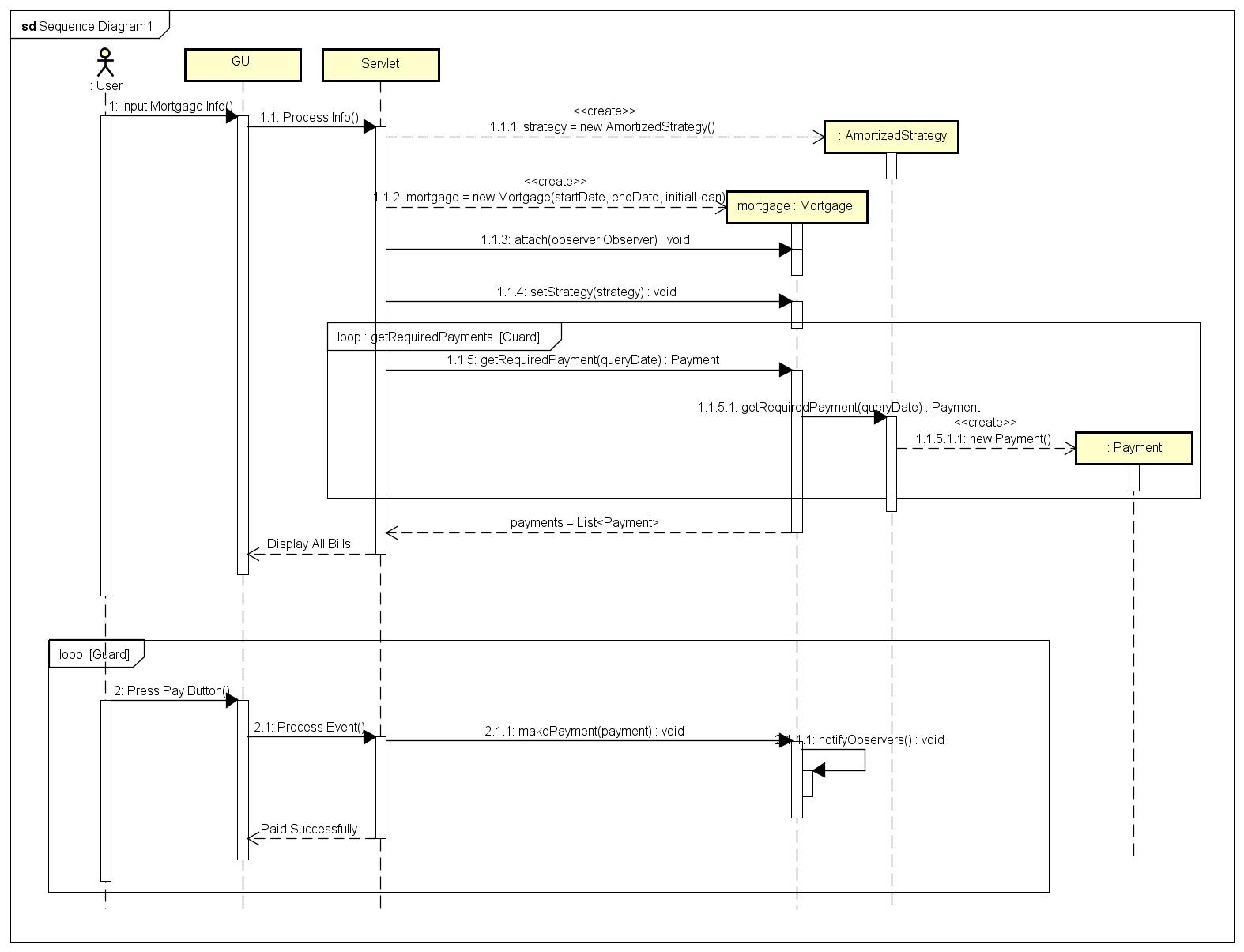
For GUI design, we use simple servlets and JSP pages. A user creates a Mortgage object using the input information, and the Mortgage object will be stored in the HTTPSession object, so the user can continue to make a payment of this mortgage.

## 2.3 Class Diagram



## 2.4 Sequence Diagram

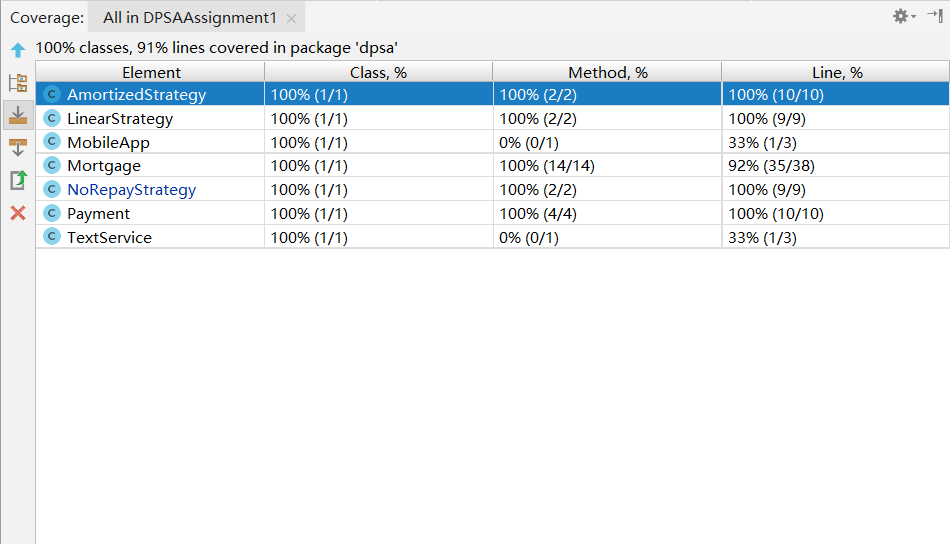
In the sequence diagram below, a user first inputs the mortgage information (start date, duration, loan, monthly interest rate, and strategy) on a JSP page, and all required payments will be displayed, and then he can make one or more payments.



# 3. Unit Testing

All tests are passed before the report is handed in.

Code coverage is shown below.



# 4. Deployment

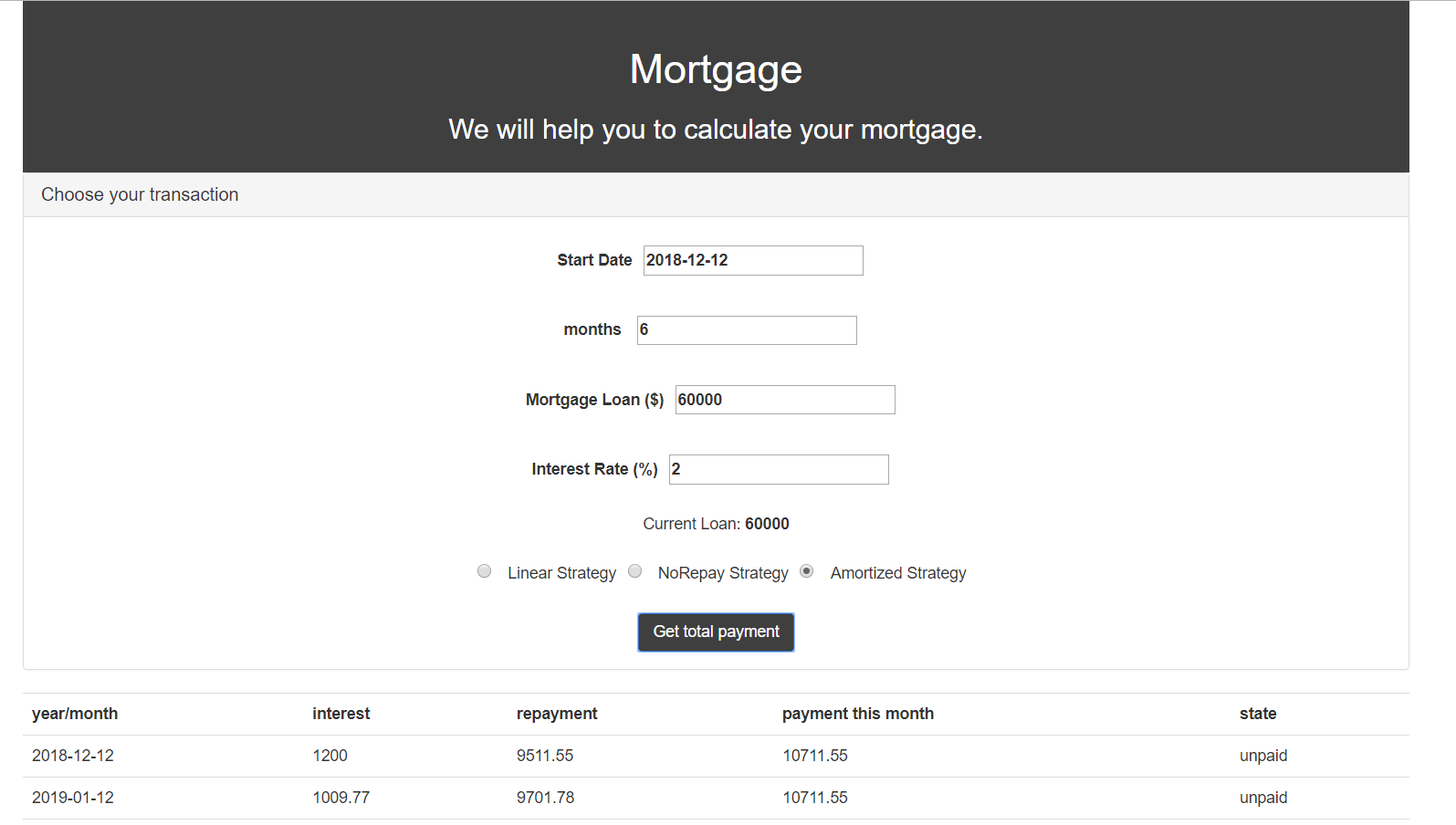
This project is built by Maven and tested only on Chrome. It should be deployed on a Tomcat server. JDK is specified as 1.8.

A .war file is provided, and the address is:

<http://localhost:8080/DPSAAssignment1-1.0-SNAPSHOT/index.jsp>

# 5. Screenshots of GUI

## 5.1 Calculate mortgage



## 5.2 Make a payment

