**Phase 1 Report**

**CMSC 495 6381 Current Trends and Projects in Computer Science (2208)**

**Group Charlie**

James Lisle

Aniebiet Jacob

Mark Tasker

Catherine Wingfield

Sam Shanzhong Yuan

***University of Maryland University College***

**Loan Amortization Calculator**

Revision: 1.1

Date: 09/20/2020

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision** | **Date** | **Name(s) of Editor** | **Description** |
| 1.0 | 09/20/2020 | Lisle | Original Document |
| 1.1 | 9/22/2020 | Lisle | Updates for client, service, source control |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Phase 1 Report**

# Project Status

The development effort for Phase 1 of the Loan Amortization Calculator for Group Charlie started on 9/16 and completed on 9/22, see table 1. There have been no significant issues that have caused any significant rethinking of loan amortization calculator design but one element that has become apparent regarding risk involves the service. If it is down any active clients will not know it until their next scheduled ping which could be as long as 15 minutes depending on when they last issued a service verification call and the service was available. Given this risk, if there is time we may consider a solution that would implement some type of notification from the service to the client.

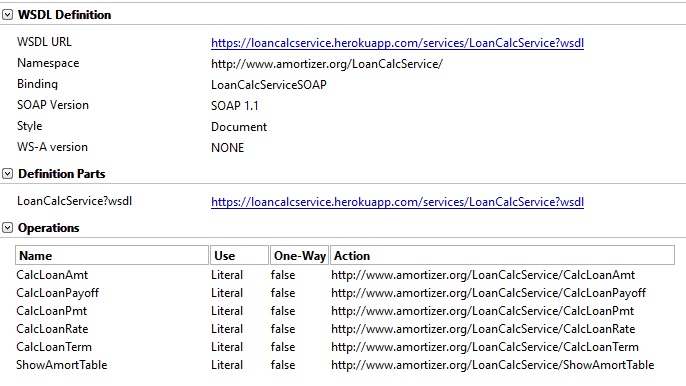
|  |  |  |
| --- | --- | --- |
| **Task** | **Date** | **Document Updates**   1. Task detail has been added to the Project Plan for development phases. |
| Phase 1 |  |
| Finalize Loan Calculator  WSDL | 9/16-9/22 |
| Create Client Project | 9/16-9/22 |
| Create Client Controller | 9/16-9/22 |
| Create Client View Class | 9/16-9/22 |
| Create Client Model Class | 9/16-9/22 |
| Create Client Stub Classes from WSDL | 9/16-9/22 |
| Check-in code to github | 9/16-9/22 |
| Create Service Project | 9/16-9/22 |
| Generate Service Classes from WSDL | 9/16-9/22 |
| Create LoanCalcEngine Class | 9/16-9/22 |
| Modify Service Skeleton to call each operation defined in the WSDL | 9/16-9/22 |
| Check-in code to github | 9/16-9/22 |
| Create Heroku Environment for Service | 9/16-9/22 |

**Table 1 – Task Detail Group Charlie in Phase 1**

**Preliminary Activities**

This application is based on a SOAP interface specification described by the Web Service Description Language or WSDL which is a contract that defines the available operations and associated request and response messages expected and returned by each, as well as a schema which defines all elements and attributes used to create request and response messages. Once the WSDL is defined, verified, and validated the service and client projects can be built. The service for this application has been built using the Apache Axis2 Web Service Project which provides an easy mechanism for packaging and deploying a service in a servlet container like Tomcat or Jetty as either a war or aar file. Since this is going to be a Cloud-based application deployed to the Heroku environment Tomcat was selected for the container.

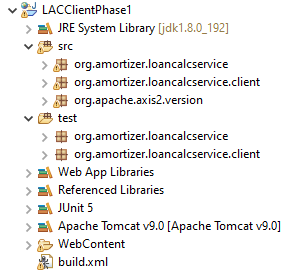
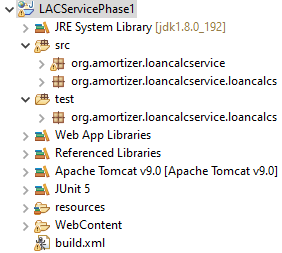
The WSDL definition for the Loan Amortization Calculator service is defined in figure 1 below.



**Figure 1 – Loan Amortization Calculator service**

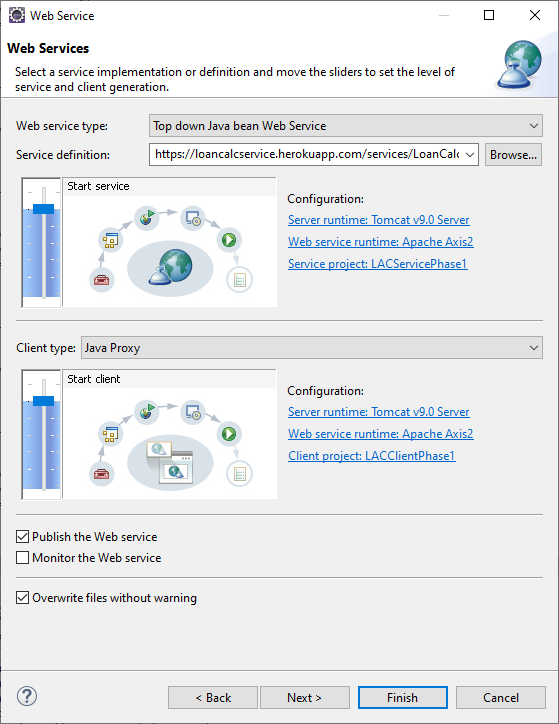
**Project Creation**

A Java Project was used for both Client and Service projects in the Eclipse IDE using Apache Axis2, Dynamic Web Module, and Java facets.

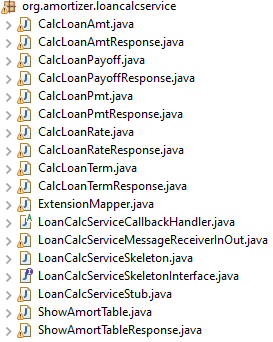
**Figure 2 – Client and Service Projects for Loan Amortization Calculator**

To generate the required stub, skeleton, and SOAP messages for this service the Apache Axis2 wsdl2Java plugin was used.



**Figure 3 – Web Service Artifact Generation Plugin Dialog in Eclipse**

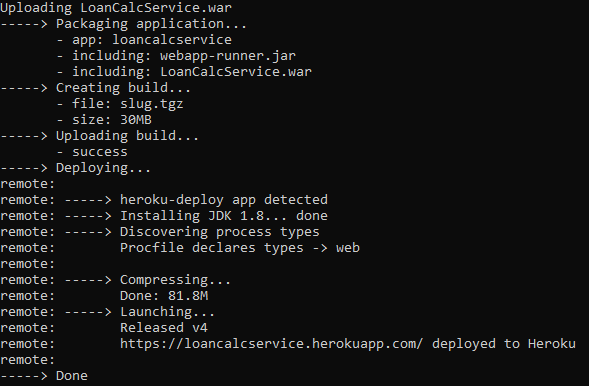
After generating the classes for the SOAP connectivity and messaging functionality, they are ready to be used on the service side to wrap the Loan Calculation Engine class and on the client side to make the call for each defined operation in the calculator service.



**Figure 4 – Generated classes from Loan Amortization Calculator WSDL.**

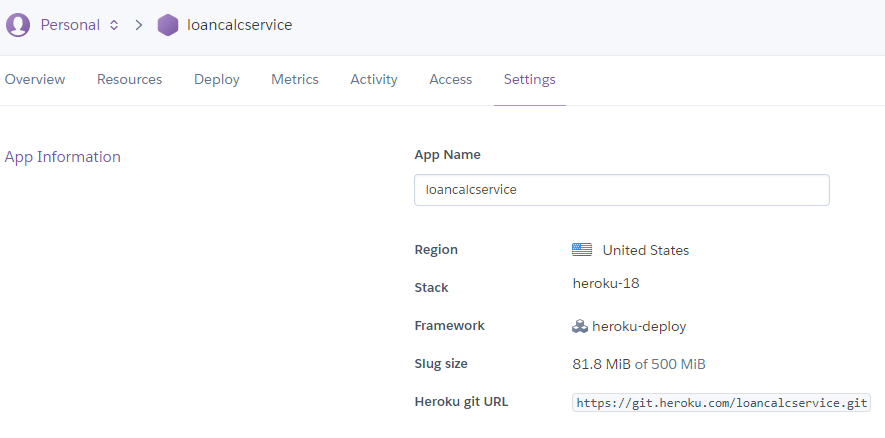
**Loan Amortization Calculator Service Phase 1 Development**

The calculation engine is defined by the amortization formula defined in the Technical Document submitted last week and consists of 4 mathematical calculations and an approximation for interest. There is also an operation for generating an amortization schedule. Currently the four standard amortization calculations have been code but the payoff calculation and the amortization schedule are targeted for phase 2. For this service a war file is being generated from the Eclipse project and deployed locally for testing and then remotely to the Heroku Cloud environment upon a successful local test. Local deployments are being performed from within Eclipse and managed by the Tomcat plugin. Remote deployments can be performed in a number of ways but for now are being executed from the Heroku CLI, see figure 5.



**Figure 5 – Deployment of Loan Amortization Calculator Service to the Heroku Environment**

Once deployed the service can be viewed from a dashboard as shown below in Figure 6.

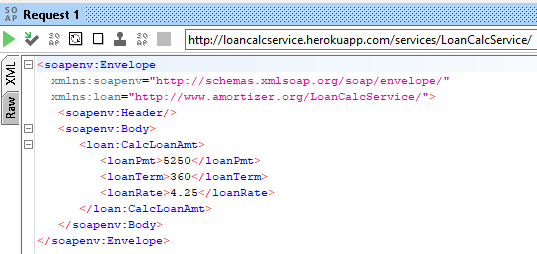


**Figure 6 – Heroku Dashboard for the Loan Amortization Calculator Service**

**Testing the Service**

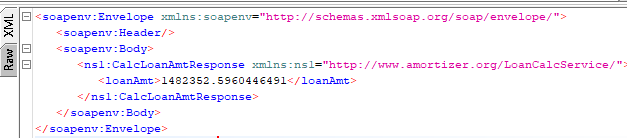
For Phase 1, tests are being submitted to the service using SoapUI as shown below.

To test the service a SOAP request message is submitted to the service, see Figure 7.



**Figure 7 – Test case for calculating loan amount**

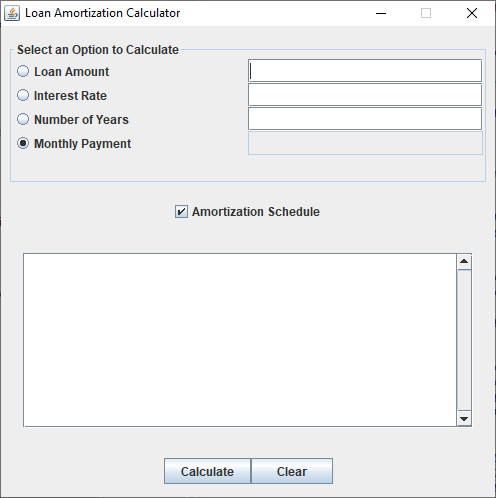
The corresponding response is shown in Figure 8.



**Figure 8 – Response generated for calculating loan amount**

**Loan Amortization Calculator Client Phase 1 Development**

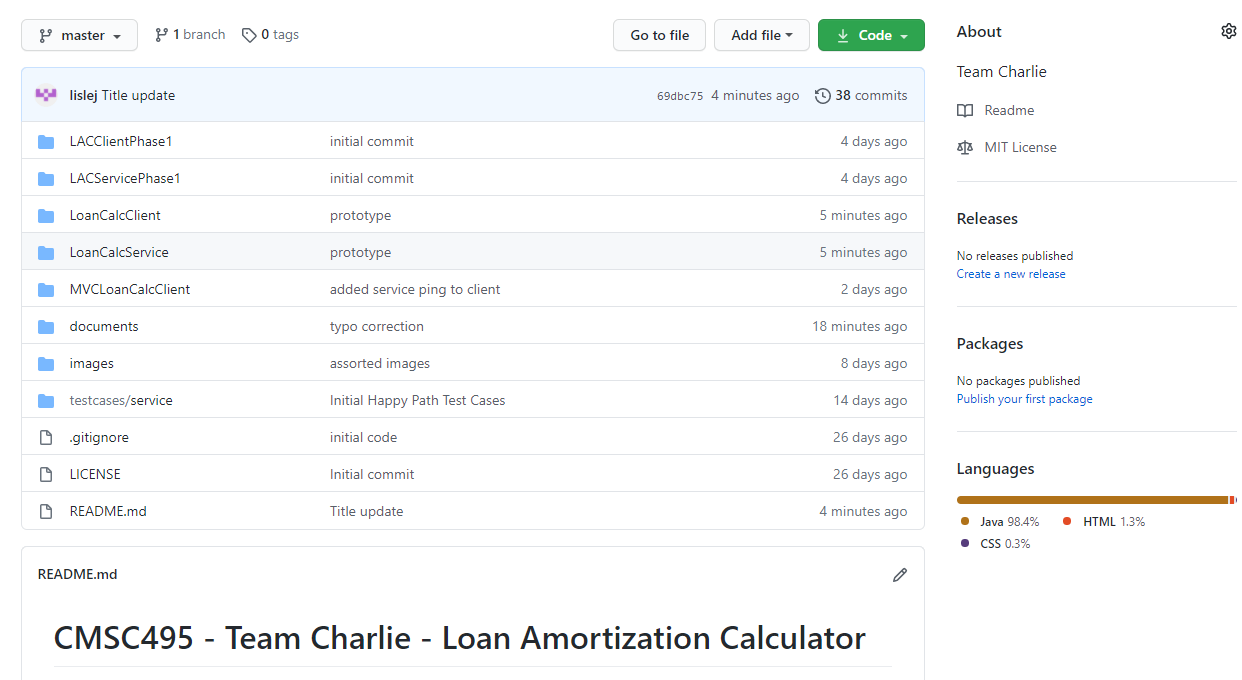
The development for the client this week consisted of a controller, a model, and a view which make up the basic client portion of the calculator, see Figure 9. The GUI will be subject to change over the phased development cycle based on developer tastes and features added.



**Figure 9 – Loan Amortization Calculator GUI**

**Loan Amortization Calculator Source Control**

This project is being developed using github source control as noted in Figure 10 below.



**Figure 10 – github repository for the Loan Amortization Calculator Project**