***SOFTWARE PROJECT MANAGEMENT PLAN (SPMP)***

**1.0 Introduction**

This section provides an overview of the software engineering project.

* 1. **Problem statement**

The purpose of the new hotel booking system is to replace the old and outdated hotel booking system. The old system is outdated, slow, inefficient and the employees have to put the customers on hold. The employees also have to work with multiple screens to pull up customer information about the reservation in order to cancel or rebook.

The new system will allow for a smooth experience for both the customers and employees. The new system will allow for outside bookings and allow the implementation of a reward system. Also employees will be able to view metrics of various customers

**1.2 Project scope**

**1.2.1 Inclusions**

Inputs

* Booking Dates, Times
* Hotel Location
* Customer Information (e.g. name, phone, address)
* Room Preferences

Processing functionality

* Checking room availability
* Processing bookings
* Calculating metrics (customer tracker)

Outputs

* Metrics: popular booked locations, revenue, rooms most booked
* Customer reward points
* Employees review reservations and/or cancel
* Employees review summary reports

**1.2.2 Exclusions**

* n/a

**1.3 Major software functions**

* User log-in management
  + Store username in database
  + Store password in a secure manner in database
  + Give users a unique ID to track throughout the system
* Administration panel for users (guests and administrators)
  + Check guest information
  + Check room availability
  + Make changes to reservations
* Sign-up for new users
  + Creating a new user ID
  + Store password in a secure manner in database
  + Store user email for security
* Track per-user metrics for bookings and number of visits
  + Use the user’s unique ID
* Track metrics/Summary Reports
  + Most booked Rooms
  + Most booked hotels
  + Track number of reward points
  + Most Visited Customers.
  + Revenue from each stay

**1.4 Performance/Behavior constraints**

* Performance:
  + Load times for any page of information should not take any longer than five (5) seconds
* Behavior:
  + Guest users should not be able to manage other guest users, only administrators should have this privilege
  + Guests should not be able to view metrics

**1.5 Management and technical constraints**

**1.5.1 Management constraints**

* Drop dead delivery date: August 19, 2020
* Lots of other courses and work. (Time management)
* Collaborating remotely
* Limited Meet time as a group

**1.5.2 Technical constraints**

N/A

**2.0 Project Estimates**

This section provides cost, effort and time estimates for the projects

**2.1 Historical data used for estimates**

Describes the historical data that is relevant to the estimates presented. This is, what historical work can be used as the basis for your estimations (e.g. prior classwork projects, work projects)

* Peter
  + Deployment of upgraded OS projects
  + Created and developed a full game of euchre, including bots using C++
  + developed a machine learning algorithm to predict forum posts using C++
  + Developed Yahtzee game using C#
  + created database management system using winforms in Visual Studio
* Monier
  + Endurance Run Management Projects (Bosch)
  + Mitsubishi ER project testing Gen5 Radar (Bosch)
  + Developed business plan for QuickPark parking indication system
* Khalid
  + P703 cluster HMI project C++ (Denso)
  + Built an 8-bit Microprocessor using VHDL. (School)
  + Developing a security system using facial recognition and Vitis AI Software.
* Jeffrey
  + Health Decisions, Inc
    - Created a customer-facing web portal, upgraded and maintained existing databases. Developed in C#, MS SQL, and created HTML / CSHTML pages with custom CSS styling
  + Autodesk, Inc.
    - Developing for desktop Data Management and Project Lifecycle application (Autodesk Vault), working with C# and C++ with Common Language Infrastructure (CLI) to interop between them, as well as MS SQL

**2.2 Initial Estimate**

Based solely on 1.0 and 2.1 details provided.

**2.2.1 SPMP Completion Estimate**

* 30 Hours

**2.2.2 Overall project estimate**

**2.2.2.1 Line-of Code Estimate**

* 1000 lines of code

**2.2.2.2 Function Estimate**

* 15

**2.2.2.3 Tasks Estimate**

What are high level tasks to complete (this is not a project plan) but tasks to think about for this estimate

* Filling out Software project management plan
* Use case
* completing and constructing a complete ER diagram
* Use case summary
* Analysis/Requirements gathering
* Building and testing software

**2.2.2.4 Total overall project time estimate in hours of effort**

200 hours

 **2.3 Estimation techniques applied and results**

A description of each estimation technique and the resultant estimates are presented here. **DO NOT COMPLETE THIS SECTION UNTIL REMAINDER OF SPMP IS COMPLETED AND REVIEWED!**

**2.3.1 Estimation technique 1 – lines of code**

estimated total lines of code/ estimated number of hours = lines of code per hour

**2.3.2 Estimate for technique 1 – lines of code**

1000/200 = 5 lines of code per hour, 200 hours

**2.3.3 Estimation technique 2 – function points**

|  |  |  |
| --- | --- | --- |
| Category | How Many? | Complexity (1-10) |
| External inputs (EI) | Customer:   1. Booking Dates 2. Hotel Location 3. user name 4. password 5. name 6. address 7. phone #   Employee:   1. booking# 2. username 3. password 4. cancel   User:   1. Is Admin?   Total: 12 | 3 |
| External outputs (EO) | Customer:   1. Availability, 2. Booking # 3. Name 4. Phone # 5. Booking Dates 6. Hotel room 7. Activity logging   Hotel:   1. Availability 2. Metrics 3. Reservations 4. Booking# 5. Hotel 6. Location of customer 7. Summary Reports 8. Most visited customers 9. Reward Points   Total: 15 | 9 |
| External Inquires (EQ) | 1. Customer username 2. Customer address 3. Customer phone # 4. Booking Date 5. Booking # 6. Hotel Room # 7. # of times customer logged in 8. # of times a room was booked 9. Username taken? 10. Is user admin? 11. Reward Points | 8 |
| Internal Logical Files (ILF) | Hotels   1. Flat file of hotels, ratings, locations 2. Table of users 3. Table of user metrics 4. Table of user rewards 5. Table of reservations | 5 |

**2.3.4 Estimate for technique 2 – function points**

12\*3+15\*9+11\*8+ 5\*5 = 36+135+88+25=284 function points

3 hours per point

284\*3 = 852 hours

**2.3.5 Estimation technique 3 – process/task**

Tables or equations associated with lines of code estimation technique m are presented. Re-estimate effort based on actual tasks defined in completed SPMP.

1. Project Approval Process
   1. Create Project Charter Task-
   2. Create SPMP task
   3. Create Use Case Summary
   4. Create Use Cases
2. Analysis Artifacts Task
3. Complete Building Application Task
4. Testing Process
   1. Complete Unit Testing Task
   2. Complete System Testing Task
5. Release Process
   1. QA Task
   2. Presentation Task

**2.3.6 Estimate for technique 3 – process/task**

400

**2.4 Reconciled Estimate**

For final time we took an average of the times calculated in the three different techniques, giving an estimated time for the project duration

Regarding cost, we took the salary of a software engineer on our team, multiplied by the number of team members, times the estimated hours for the project

Final Time: 466 hours

Final Cost: $15 / hr \* 4 resources \* 466 hours = $27,960

**2.5 Project Resources**

* Khalid, Peter, Monier, Jeffrey
* Visual Studio 2017 - 2019 IDE
* Standard C++ libraries
* SQL Databases
* Laptop/Personal Device
* Phones

**3.0 Risk Management**

This section discusses project risks and the approach to managing them.

**3.1 Project Risk Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Probability** | **Impact** | **Mitigation** | **Contingency Plan** |
| Time management | Medium | High | Split up the work, weekly check ins, work consistently | Focus on most important aspects of the project |
| Functionality | Low | High | Thorough testing as we go through. | Try to get the main components to work first. (Prioritize) |
| Losing a team member temporarily | Low | Medium-High | To know the status of our team members and using github to have access to their work. Take notes for absent team member | Temporarily reassigning work as needed depending on the length of absence of team member |

**3.2 Overview of Risk Mitigation, Monitoring, Management**

How will Risk Mitigation, Monitoring and Management be handled. Communication plan, when to apply the mitigation strategy, etc.

* Apply mitigation strategy as soon as risk appears more apparent
* Groupchats-notify if going to be absent or falling behind in project work
* Github- back our code and other documents
* Weekly Meetings- check in on project work status of each team member

**4.0 Project Schedule**

This section presents an overview of project tasks and the output of a project scheduling tool.

**4.1 Project task set**

The process model, framework activities and task set that have been selected for the project are presented in this section.

Process Model: Waterfall

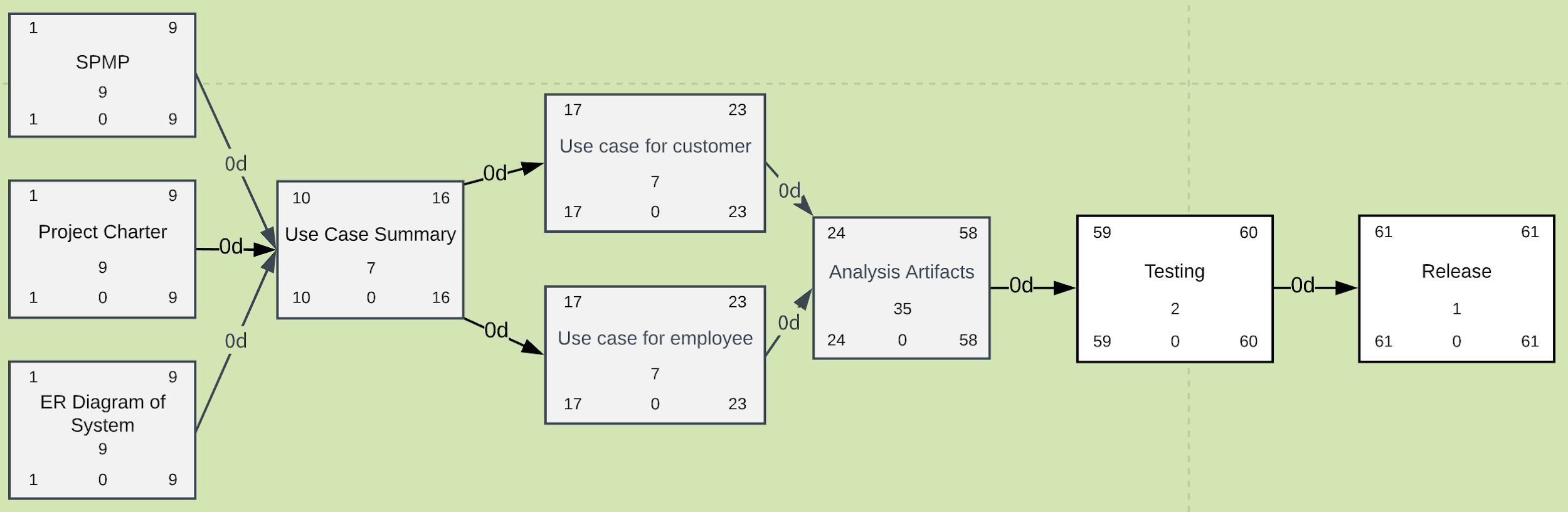
Recurring Tasks

* Weekly Team meetings
* Weekly status report
* Coordinating weekly meetings

Framework activities and Tasks:

* Create project charter
* Create SPMP
* Create Complete ERD
* Create Use Case Summary
* Create Use Cases
* Create Analysis Artifacts
* Building & Testing System
* Release To Customer
* Maintenance

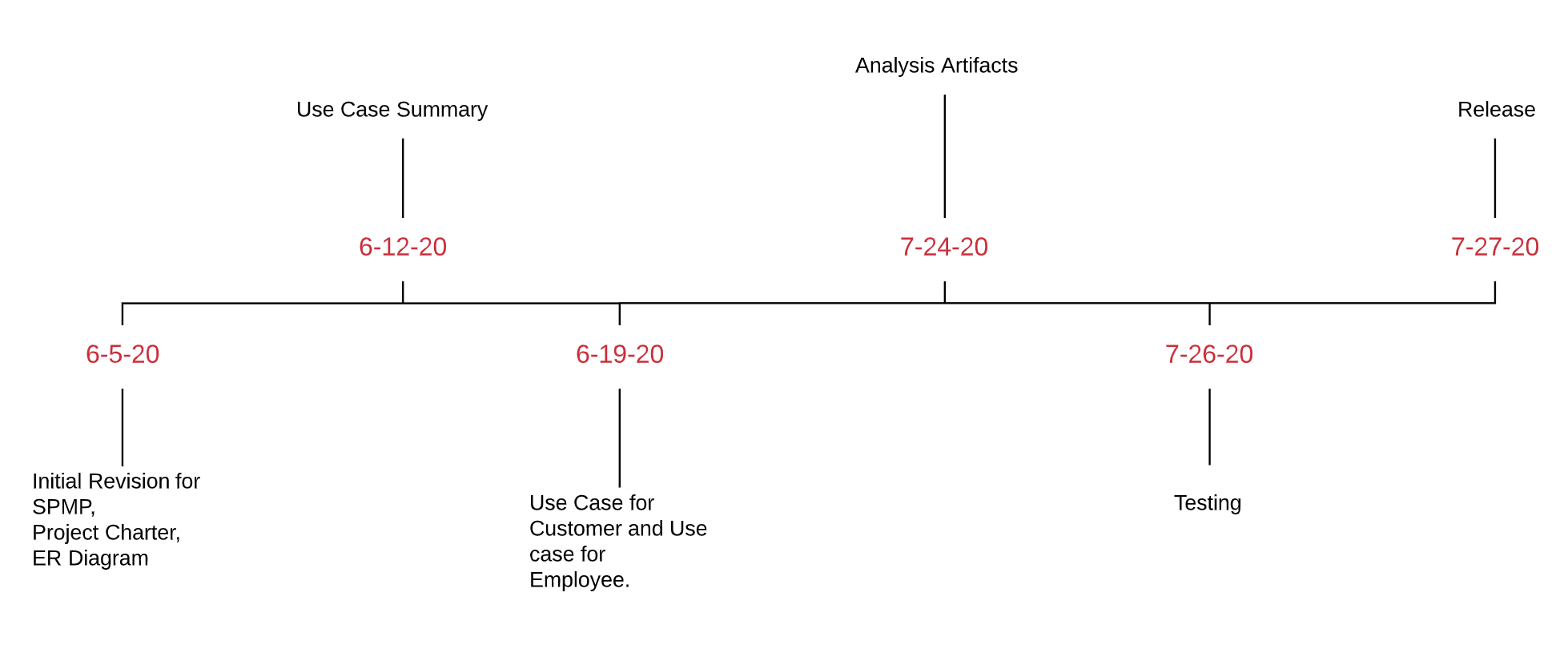
**4.2 Task network**



Calculation: Tasks start date: 5-27-2020, Tasks Presentation Date: 7-27-2020, Duration: 61 days

**4.3 Timeline chart**

A project timeline chart is presented. This may include a timeline for the entire project or for each staff member.



**5.0 Staff Organization**

The manner in which staff are organized and the mechanisms for reporting are noted.

**5.1 Team structure**

* Team Lead - Peter
* Software Developers - Khalid, Monier, Jeffrey, Peter

**5.2 Management reporting and communication**

Mechanisms for progress reporting and inter/intra team communication are identified.

* Group chats
* Weekly Team meetings
* Class meetings
* Meetings with Client (Professor Tom Steiner)

**6.0 Tracking and Control Mechanisms**

Techniques to be used for project tracking and control are identified.

**6.1 Quality assurance and control**

An overview of SQA activities is provided. This is an outline at this point and will be used to create a detailed plan later in the project.

* Unit Testing
* System-Wide Testing
  + User Accounts
  + Administration

**6.2 Change management and control**

An overview of SCM activities is provided on how changes will be handled – communication of a change, how decisions made of approval, defer or reject change request.

* Communication of additions via Group Chat
* Coordination and approval of merging branches into master
* Commits on Github

**6.3 Tools**

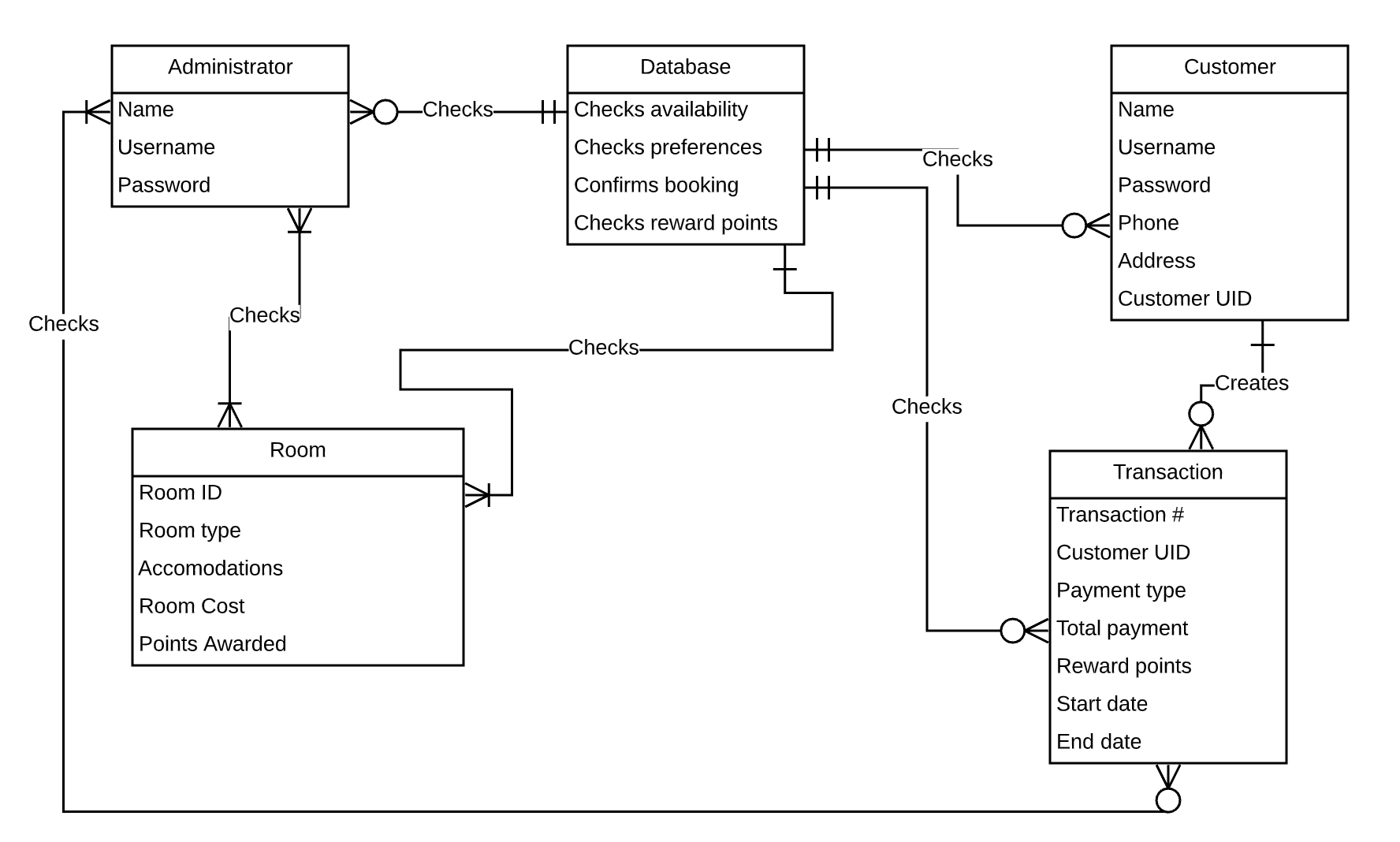
What tools will be used to control access and versioning of artifacts?

* Github
* Google Docs

**7.0 Appendix**

Supplementary information is provided here.

**ER Diagram**

****