Amazonian Prime

Terms of Reference

April. 4, 2023
Version 2.0

Document Information

Revision History

Date	Version	Status	Prepared by	Comments
Jan 24, 2023	1.0	Graded	Amazonian Prime	
Feb 11, 2023	2.0	Complete	Amazonian Prime	Revisions for Midterm Check-in: • Defined BSD in glossary • Explained our choice for BSD 2-clause • Updated Step Functions, Lambda and S3 estimated pricing breakdown • Added more technical dependencies • Revised project timelines • Elaborated on risk reasoning
April 3, 2023	2.0	Complete	Amazonian Prime	No revisions necessary for final submission

Document Control

Role	Name	E-mail	Telephone
Professor	Jerry Jim		
TA	Marie Salomon		
TA	Shijun Shen		
Project Sponsor	Peter Smith		
	Mahmoud Al Khatib	mahmoudalkhatib.ubc@gmail.com	
	Michael He	michaelhe17@gmail.com	
	Joshua Luong	joshualuong@hotmail.com	
	Tristan Martinuson	tmartinuson@gmail.com	
	Elaine Shi	elaineshi328@gmail.com	
	William Suryawidjaja	suryawidjajaw@gmail.com	

Approval

Role	Name	Signature	Sign-off Date
Amazonian Prime Member	Mahmoud Al Khatib	Mahmoud Al Khatib	April 4, 2023
Amazonian Prime Member	Michael He	Michael He	April 4, 2023
Amazonian Prime Member	Joshua Luong	Joshua Luong	April 4, 2023
Amazonian Prime Member	Tristan Martinuson	Tristan Martinuson	April 4, 2023
Amazonian Prime Member	Elaine Shi	Elaine Shi	April 4, 2023
Amazonian Prime Member	William Suryawidjaja	William Suryawidjaja	April 4, 2023

Table of Contents

Introduction

We are a team of 6 students at the University of British Columbia completing our upper years in Computer Science. We bring experience from various companies in full-stack development, and we have the pleasure to be working with Amazon Web Services (AWS) as Team Amazonian Prime via CPSC 319 (Software Engineering Project).

Team Amazonian Prime has been tasked to develop an AWS-based SaaS solution to help drive internal staff/ Amazonian engagement by introducing an online buy and sell platform. Specifically, due to the new norm of remote work, we aim to enhance relationships and interactions that allow for virtual and physical connection among Amazonians. Our solution will also have an inherent environmental benefit by providing a system to incentivize the recycling and reusing of products, thereby reducing waste.

Rather than a simple buy-and-sell platform, we hope that staff members can participate as buyers and sellers within their local Amazon office, whereby items can be sold, shipped, and/or traded. In addition, we will offer administration roles that will allow admins to add/remove users, delete postings, and delete orders. The new e-commerce site will also offer to allow Amazon HR to sell staff merchandise (e.g. staff discounted logo merchandise, free merchandise, etc.) The site will also have basic and classic functions for e-commerce sites such as the ability to add/ remove products in carts. Pending available time (with an aim to have project completion by April 3rd), the e-commerce site will also allow for order cancellation, create a mobile-friendly web version, offer reports for sellers of their inventory and products sold, etc.

Amazonian Prime's Terms of Reference will address project objectives, where we will identify resource requirements (including budgets, team members, etc.), and project risks, and offer a high-level project delivery plan. This is intended to be a high-level document which will house our stakeholder's agreements and expectations throughout the project life cycle.

Project Information

Project Overview

Background Information

Amazon Web Services team in Vancouver aims to improve the office atmosphere and foster a closer connection among employees. We aim to create an internal commerce platform where employees can buy and sell physical items with one another.

Problem/Opportunity Statement

The AWS Vancouver team is facing a challenge in fostering staff engagement and building stronger relationships among employees, especially with the new norm of remote working. To address this challenge, the team is proposing to develop an AWS-based SaaS solution that allows for virtual and physical connections among staff through an internal commerce marketplace. The proposed solution aims to improve staff networking and engagement and to make the most of the AWS technologies and solutions.

Goals and Objectives

- Develop an AWS-based SaaS internal commerce marketplace where staff can sell/trade physical items with other staff members, in order to improve engagement.

Deliverables

- Employee buyers module in which potential buyers can:
    - Register their information into the system (name, shipping location, payment details)

    - View their order history and shipping status
- Employee sellers module in which potential sellers can:
    - Register their information into the system (name, banking information)
    - Post their products with information such as a description, price, quantity, and pictures of the product
- Administration module in which administrators can:
    - Add or remove users, postings, orders
- Marketplace system which can:
    - Search for a product by name or category
    - Place products into a shopping cart
    - Calculate shipping costs and taxes
    - Simulate payment
    - Simulate shipping notification and details
- A relational database in order to manage user, item, and order information
- A simple authentication system (Google OAuth is fine)

Benefits

- Increase engagement in the workplace, especially amongst remote employees
- Increase interactions that allow for virtual and physical connection amongst staff
- Foster networking between employees and different departments
- Reduce waste by offering a second life for unwanted/ unneeded items

Scope

In Scope

- AWS-based SaaS e-commerce marketplace system for AWS staff
    - Employee buyers, sellers, and administrative module
- Multi-device and multi-browser standard
    - Chrome, Safari, and Firefox
- Performance
    - Must support a user base of 500
    - Searches and filter time must be less than 4 seconds
- Concurrency
    - Support up to 10 concurrent users
- Authentication
    - Use of Google OAuth is acceptable

Out-of-Scope

- Implementation of shipping and payment systems

- Mobile-friendly web version of the application
- Geography-based tax and shipping cost calculation

Sustainability Impacts

Our project has a direct impact on sustainability by facilitating the recycling and reusing of products. From a long-term perspective, increased engagement in the workplace amongst remote employees can mitigate the need for trips to the office or other physical networking events in which employees will need to travel.

Open Source Licensing

As our project will most likely be used as a pilot/proof-of-concept that will likely be rewritten, we have chosen to use a simple permissive license, BSD 2-Clause “Simplified” License, as our Open Source License. We have chosen BSD 2-clause specifically as it is less restrictive than the commonly used BSD 3-clause, which has the additional clause:

> “Neither the name of the copyright holder nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.”

Which may prove to be restrictive if our project is used as is and not rewritten.

Project Constraints, Assumptions, Dependencies and Risks

Constraints

- Limited amount of time to plan and meet all required deliverables
    - ~8 weeks for planning, ~4 weeks for implementation, 1 week warranty period; overall ~13 weeks to build a service that meets all requirements outlined by sponsor
- Limited budget to meet all deliverables
    - AWS student credits
    - CPSC 319 budget
- Limited amount of resources, particularly human resources
    - Limited availability of sponsor to answer questions pertaining to requirements
    - Limited availability of TA to offer guidance and answer questions pertaining to the project
    - Limited availability and capacity of team members to meet, collaborate, and work on project
    - Limited expertise and knowledge. Team members do not have extensive experience working with AWS or building e-commerce platforms.
    - Limited technical resources. Team members all have different devices with different capacities.

Assumptions

- Maximum user base of 500
- Maximum 10 concurrent users
- Average of 3 postings per month per user
- Average of 5 images per posting
- Average image size of 5MB
- Average of 200 characters for each posting description

Dependencies

- Our technology stack will be React/ Redux (frontend), Material UI, Node JS (backend), Aurora MySQL (database)
- Google OAuth, AWS Technologies (Step Functions, Lambda, S3, CloudFormation, Simple Email Service, API Gateway, Amplify)
- The backend code will be executed using Lambda functions through calls to the API Gateway
- Sponsor (requirements, availability, expectations)

Summary Risk Assessment

Risk ID	Reasoning	Assessment
1	There's a risk that members encounter issues while learning AWS that causes a stretch in timeline	Factor in the cost of learning when planning, have group learning sessions
2	There's a risk that the sponsor modifies the request which can cause unplanned changes to our system	Make adjustments to the current plan as soon as possible
3	There's a risk that our estimation of time and difficulty of tasks are different than the actual, which will cause the inability to deliver the product on time	Aim for an executable product before any advanced functions
4	There's a risk that the newly formed team has difficulty maintaining organization that causes the project to have poor structure	Have frequent meetings and encourage communication
5	There's a risk that team members are not always able to attend every discussion session, and the catch up time may cause delaying the project past due date	Plan ahead of time or create set schedules
6	There's a risk of implementing a feature that fails to match the sponsor's vision that causes the delivery of an inaccurate product	Be careful in following the project document and verify design with the managers (TA) frequently
7	There's a risk that the sponsor requests revision on the design or implementation on the product that causes unplanned changes	Verify design with sponsor as often as we can

Find the associated Risk IDs in the Appendix.

Project Approach and Acceptance Criteria

Project Approach

We will be using a waterfall release approach and collecting requirements from the Project Sponsor as the project progresses. We will also communicate with Industry Stakeholders through IT Manager (Teaching Assistant).

Estimated Schedule

Milestone or Key Activity	Start	Complete
Initiation	Jan 13	Jan 17
Requirements	Jan 18	Feb 1
Environment Setup + Source Control	Feb 2	Feb 7

Demo Implementation	Feb 8	Feb 14
Midterm Presentation	Feb 15	Feb 15
MVP Implementation + Feedback	Feb 16	Mar 15
Project Completion	Mar 16	Apr 5
Warranty Period	Apr 6	Apr 20

Estimated Effort and Cost

There will only be two estimated costs, which is the cost of using the database and the cost of using AWS services. See the Appendix for detailed calculations.

- We estimate an average of 37.5 GB to be stored in our database each month, and at the cost of $0.10/GB, this works out to $3.75 per month for the database (S3)
- We estimate an average of 2000 function requests per each user per month, resulting in a total of 1,000,000 function calls per month at a cost of $1.45 for AWS Lambda. Additional 1,000,000 requests cost $0.00375 thereafter according to AWS.
- We estimate a maximum of 100 state transitions per 500 users per month, resulting in a cost of $0.00025 for AWS Step Functions

These calculations are the upper bound on the amount of cost we are estimating for the development of the project. However, because our application usage should be low, the AWS Free Tier of the services will allow us to develop our application at low to no cost.

Acceptance Criteria

Refer to the table in the Appendix.

Project Governance

Project Team Responsibilities

Although we will all collaborate and work on different parts of the project together, each team member will be responsible for one or two individual aspects of the project.

Responsibility	Owner(s)
Backend/ DevOps: Lambda	Tristan
Backend/ DevOps: Step Functions	William
Backend/ DevOps: Aurora RDS	Mahmoud
Backend/ DevOps: API Gateway	Elaine
Frontend (React/ Redux in TypeScript?)	Josh, Michael
Design (Figma mockup)	Josh
Testing/ QA	Michael
Project Manager	Josh - Rotate based on milestone

Project Role	Person Responsible	Contact Details
Project Sponsor	Peter Smith	Through Marie or Shijun
Budget Owner(s)	Teaching Team	Through Canvas

Responsibility Matrix

Refer to the table in the appendix.

Project Communication Plan

Project Meetings

We will meet virtually every Monday to collaborate on the project as well as meet in person every Wednesday. We will also communicate through a discord channel on a moment-to-moment basis.

Project Reporting

We will report to the teaching team every Wednesday as well as the sponsor on several specified dates and to our Project Manager on Monday syncs.

Appendix

Glossary of Terms / Abbreviations

Terms / Abbreviations	Description
SaaS	Software as a Service, a method of software delivery and licensing in which software is accessed online via a subscription
React/Redux	React is a front-end JavaScript library for building user interfaces managed and based on modular components. Redux is a JavaScript library for managing and centralizing application state used in conjunction with React..
Lambda	Serverless compute service that allows code to be hosted and run without provisioning or managing servers. It automatically scales and monitors with the application.
Step Function	Service that enables coordination and initiation of distributed workflows and applications through the use of visual workflow diagrams
Aurora RDS	Relational database service provided by Amazon Web Services (AWS) designed to provide high availability, performance, and scalability for cloud-based applications
Google OAuth	Provides user authentication with a service without the need of maintaining user private information such as usernames, passwords and other sensitive details.
BSD	Berkley Software Distribution, a specific type of open source license that does not put requirements on redistribution

Risk ID Table

Risk ID	Risk Description
1	Difficulty in learning AWS and other new technologies
2	Change in specification or goal from the sponsor
3	Error in the estimation of timeline due to setting too ambitious goals
4	Internal issues of a newly formed team
5	Delays or miscommunication between members due to schedule differences
6	Incorrect design or implementation due to incorrect understanding of specs
7	Delay in progress when the product does not get accepted by the sponsor in one iteration

Acceptance Criteria

Criterion (Given)	How Measured (Then)	When Measured
Buyer provides personal information	Buyer’s info is stored in the system	User is accessing buyers module
Buyer searches for purchased items	All of the buyers’ activity, including order history and shipping status, is displayed	User is accessing buyers module
Seller provides personal information	Seller’s info is stored in the system	User is accessing sellers module
Seller posts an item to sell	Seller’s products are posted, along with their detailed information	User is accessing sellers module

Administrator adds/ removes an entity	The entity (user/ postings/ order) is removed from the system	Administrator is accessing the administration module
User searches for a product by name or category	Products with the specified name or category are displayed in alphabetical order	User is in the marketplace module
User places product into shopping cart	A unit of the product is added to the user's shopping cart	User is in the marketplace module
User requests a calculation of shipping costs and taxes	The total costs of shipping and taxes are displayed to the user	User is in the marketplace module
User performs a payment process	The purchase is handled and funds transferred from the buyer to the seller's account	User is in the marketplace module
User authenticates their account using Google OAuth	User enters their account if the authentication is successful	User is accessing the login page

Responsibility Matrix

Deliverable	Project Sponsor	Budget Owner	User Representative	Business Project Manager	IT Management	Project Manager	IT Staff	IT Security	IT Architecture	Subject Matter Experts	Project Audit and Review	Compliance	IT Operations	Test Manager
Detailed Requirements	I	I	A	C	I	R	C	NA	NA	C	I	I	I	NA
Implementation & Backout Plan	I	I	A	A	I	A	C	NA	NA	R	I	I	I	NA
Project Completion Certificate	A	A	A	AC	A	R	C	NA	NA	C	I	I	I	NA

R = Responsible

A = Accountable / Approves / Signs Off

C = Consults / Contributes

I – Informed or copied

Estimated Cost Calculations

AWS S3
$5\ images\ *\ 3\ posts/month\ *\ 500\ users\ =\ 7,500\ images$
$7500\ images\ *\ 5\ MB\ (average\ size\ of\ an\ image)\ =\ 37.5\ GB\ of\ storage\ for\ images$
$200\ characters\ per\ description\ *\ 3\ posts/month\ *\ 500\ users\ =\ 400,000\ characters$
$300,000\ characters\ *\ 2\ bytes\ (size\ of\ a\ char)\ =\ 600\ KB\ of\ storage\ for\ descriptions$
$Total\ Estimated\ Storage\ =\ 37.5\ GB\ +\ 600KB\ =\ 37.5\ of\ storage\ to\ hold\ images\ and\ descriptions\ for\ 500\ users$

AWS S3

AWS Step Functions
$500\ users\ *\ 30\ state\ transitions\ (updates\ to\ profile,\ search,\ buying,\ selling,\ etc)\ =\ 1,500\ state\ transitions$
$\$0.025\ per\ 1,000\ state\ transitions\ /\ month\ *\ 1,500\ state\ transitions\ =\ \$0.0375$

AWS Lambda
$500\ Users\ *\ 2,000\ estimated\ HTTP\ requests\ per\ month\ =\ 1,000,000\ function\ calls\ to\ AWS\ Lambda$
$1,000,000\ function\ calls\ per\ month\ ->\ Request\ Costs:\ \$0.20\ +\ Execution\ Costs:\ 1.25\ per\ month$
$Length\ of\ service\ required\ \sim 4\ months\ =\ 4\ *\ \$1.45\ =\ \$5.80$
$Estimated\ total\ budget\ (with\ projected\ room)\ for\ AWS\ Lambda:\ \$10.00$