Shuqa

Soran Qiji

CST-452 Capstone Project Final Architecture & Design

Grand Canyon University

Instructor: Professor Mark Reha

Revision: 4.0

Date: 4/27/2025

ABSTRACT

The Shuqa project, Shuqa is a Chaldean (Aramaic) word for bazar, is a web application that people can use to sell and purchase products locally in areas where bank cards are not widely used. Everyone can access the website with no need for credit/debit cards or any other online payment methods such as PayPal, as cash is the only accepted payment method. And buyers and sellers should arrange the location of the purchased item pick up. The users must register providing their information such as full name, email address, and phone number to create usernames to access the website. No membership or fees are required.

History and Signoff Sheet

Change Record

Date	Author	Revision Notes
11/17/2024	Soran Qiji	Initial draft for review/discussion
3/24/3035	Soran Qiji	Design Introduction (removing cloud host while the product is not hosted on cloud), Detailed High-Level Solution Design (Adding reset password), Logical Solution Design (Adding the use of Gmail to send the reset password link, update the diagram), Updating General Technical Approach, update Key Technical Design Decisions.
4/15/2025	Soran Qiji	Detailed High-Level Solution Design (Adding multiple images upload), ER diagram (change the type of images to string array), Appendix A (Risks and Issues).
4/26/2025	Soran Qiji	Physical Solution Design, final touches on Wireframes, Flowcharts, Sitemap, UML, and risks/issues.

Overall Instructor Feedback/Comments

Overall Instructor Feedback/Comments

Integrated Instructor Feedback into Project Documentation

☒ Yes ☐ No

Table of Contents

Design Introduction

This is a simple web app that is designed to be used as a local online marketplace. The customer can browse the items that are listed on the website by other local sellers without need to login. The customer, to access the website features such create a post, update a post, and delete a post, need to create an account first. The website provides a navbar that will make navigate between pages easy and smooth. Also, the web application uses a simple design for the web pages that can be easy to use and understandable by everyone. This website will not use any payment method such as credit cards or a third-party payment, that because the area that this application is design to be used in the bank cards or a third-party payment are not widely used.

Deliverable Acceptance Log

ID	Deliverable Description	Comments	Evaluator (internal or external as applicable)	Status	Date of Decision
1					
2					
3					
4					
5					

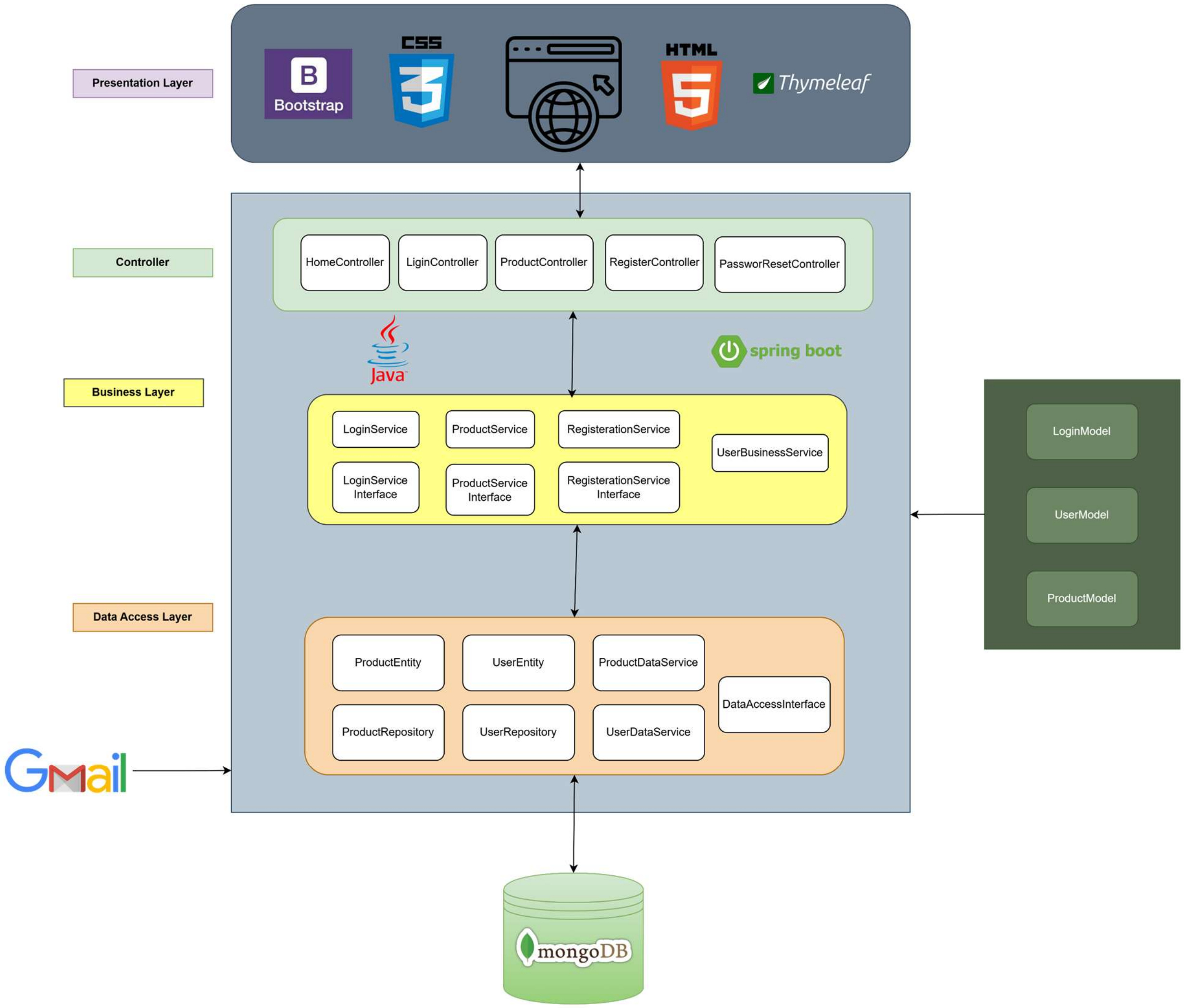
Detailed High-Level Solution Design

Proof of Concepts		
Description	**Rationale**	**Results**
1 – Create test users	To test if the user can only update and delete the item that posted by them and the My Product page only show the products that are listed by specific user	The application work and each user only can have its items and can update and delete its item only
2 – Reset Password	The user will request a link for password reset using email address, system check for email validation, then the link will be sent to email, and user can update the password through the link.	The reset password is work, it is checking for email validation, also it checks the password pattern.
3- Upload multiple images	The user can upload images for the product, the system will check the file format to make sure that only image format are being uploaded	Multiple images can be uploaded successfully and the file format is checked so that only image can be uploaded.

Hardware and Software Technologies
1 – Bootstrap (version 5.3)
2 – Thymeleaf (version 3.1)
3 – HTML and CSS
4 – Spring Boot (version 4)
5 – Java (version 17)
6 - MongoDB
7- Gmail for sending reset password link
8- VS code (version 1.95)
9 – Computer with Windows OS, CPU core I7, RAM 20 GB, storage 500 GB.

Logical Solution Design:

This is a logical system design for my project, that shows the use of MVC that separates the application logic into three interconnected parts, presentation layer, business layer, and data access layer. The user will interact with the presentation layer using devices such as personal computers or mobile devices. The web is using Bootstrap and Thymeleaf to build the application frond-end (user interface) which Bootstrap provides CSS frameworks, additional CSS for styling, and Tymeleaf provides HTML templates. The presentation layer will communicate with controllers in the business layer. The controllers are bridged between presentation layer and business layer which handle HTTP requests and responses. The business layer, which is built using Spring Boot and Java, is the layer that contains the logic and business rules of the application which is responsible for processing user requests and managing interactions between presentation layer and database via Data Access layer. Data Access is the layer that is responsible for interacting with the database to perform create, read, update, and delete (CRUD). Database (MongoDB) is used to store application data. Gmail (stmp.gmail.com) is used to send a password reset link to the user.



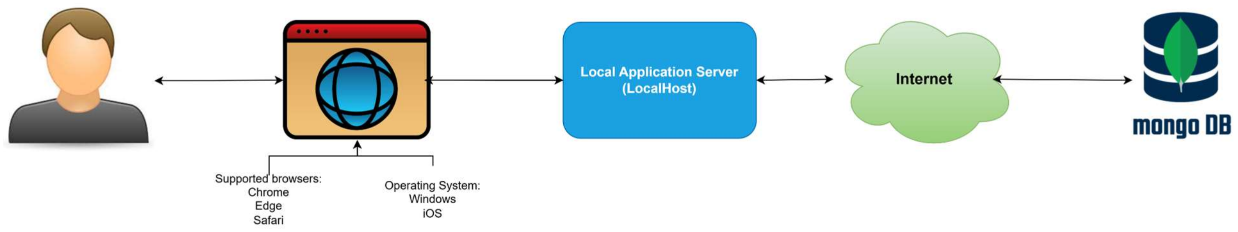
Physical Solution Design:

The Shuqa web application is currently hosted locally on a development laptop. Users can access the application through any modern web browser on a computer.

The application server runs locally, while all data is managed and stored remotely in a MongoDB Atlas database. MongoDB Atlas provides a secure, cloud-based environment for storing user accounts, product listings, and other application data.

This setup allows the application to benefit from the scalability, reliability, and backup features of MongoDB Atlas without the need for local database management. Security measures such as password hashing and secure authentication practices are applied to protect sensitive user information.

While the application is currently used for development and testing, the design can easily be scaled to a production environment by deploying the application server to a cloud hosting platform or dedicated server when needed.



Detailed Technical Design:

General Technical Approach:

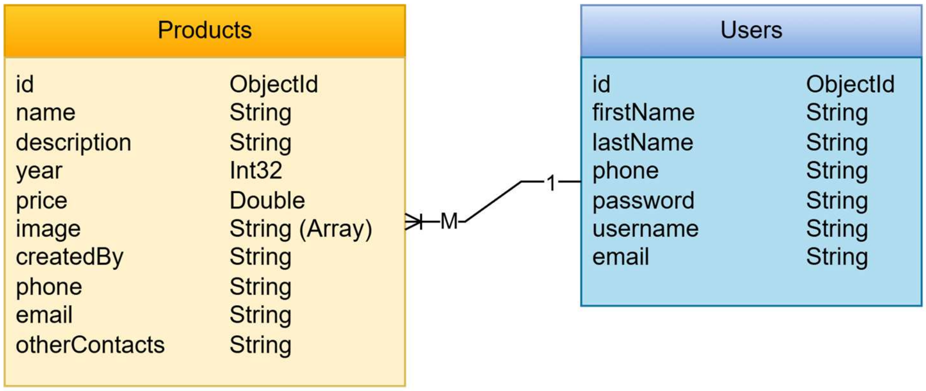
The approach that will be taken to develop this website is implementing more features in the future, such as adding a payment method in the future as the web application expands to be used in other regions of my country. Adding a feedback section that will let the user leave feedback that will help to develop and improve the website. Create an admin rule that will be able to delete any post or user, which is out of scope for right now, and add chat/messaging between buyer and seller. For now, the web application will use MongoDB as database support, the products' images are stored locally, in the feature a cloud based database will be used such as Amazon S3 or Microsoft Azure Blob. Spring boot, and bootstrap will be used as the framework and thymeleaf as java HTML template engine to develop and improve the web application for addition styling Bootstrap with CSS will be used as well.

Key Technical Design Decisions:

VS Code: will be the IDE of choice to build the and develop the web application because it supports Java, and it is easy to use.
Spring Boot: a powerful framework choice for its simplicity, strong dependency injection, simplified configuration, and it supports modern web applications.
Java: a java program can run on any platform that supports JVM. Also, java support spring frameworks. In addition, java offers a wide range of libraries.
Thymeleaf: a popular server-side Java template engine for Java web applications. Thymeleaf provides HTML templets that can be directly opened and previewed in browsers. Also, it integrates very good with spring boot.
Bootstrap: a popular front-end framework and mobile-first application, bootstrap will ensure that the website will look good on all screen sizes.
CSS: for styling.
MongoDB: MongoDB Atles is used to store the data for stability and easy access and use.
Gmail: Gmail is used to send the reset password link via email to the user.
Figma: Figma is used to design wireframes and flowcharts.

Schema ER Diagram:

MongoDB is used for this application. As the picture below shows that the database has two collections, Products that will have all the product posts that are created by user, and users' collection, which will include information about the registered user. As the diagram shows that the relationship between the users and the products is one to many which mean that each user can have many posts (product post).

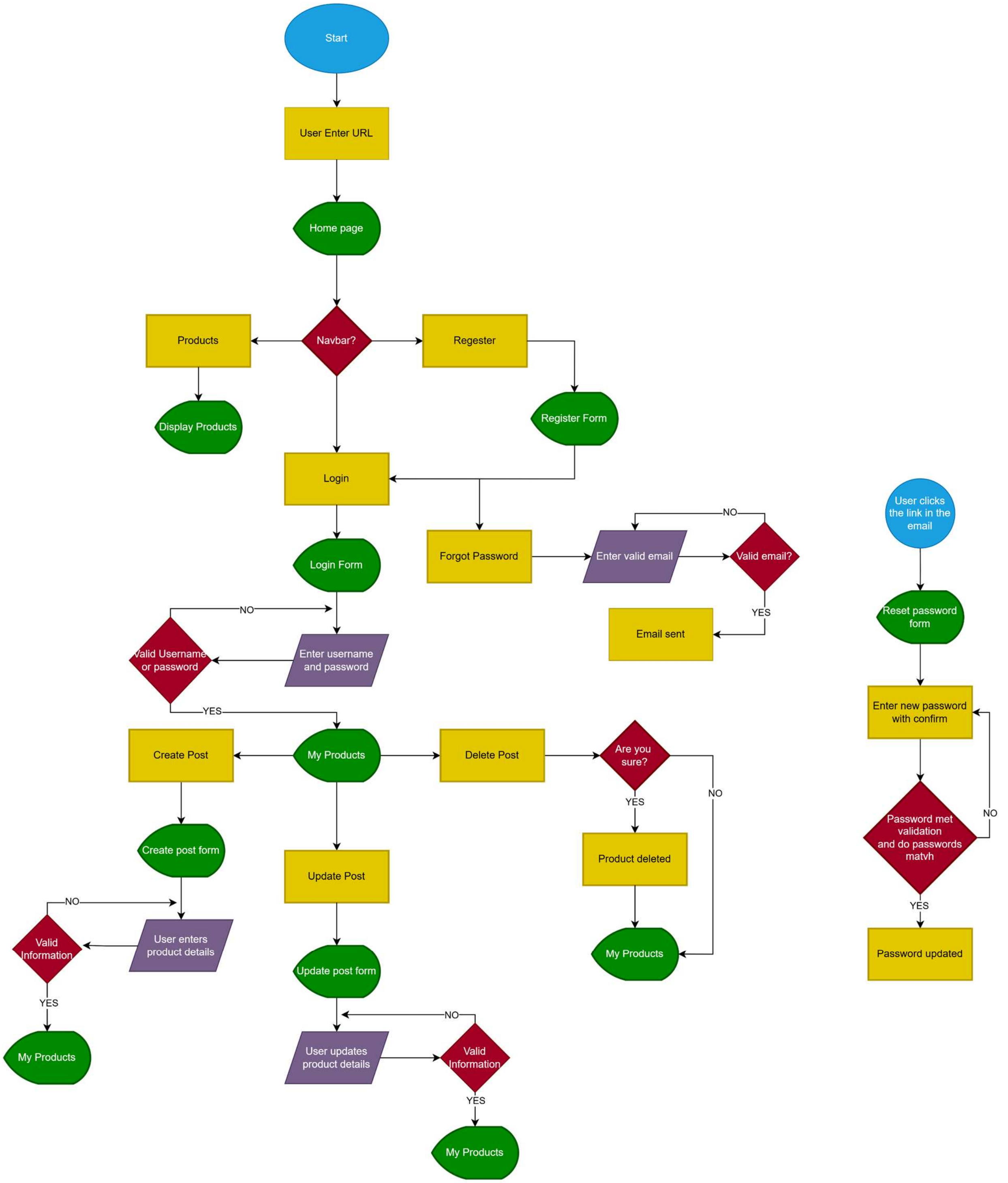


Products	
id	ObjectId
name	String
description	String
year	Int32
price	Double
image	String (Array)
createdBy	String
phone	String
email	String
otherContacts	String

Users	
id	ObjectId
firstName	String
lastName	String
phone	String
password	String
username	String
email	String

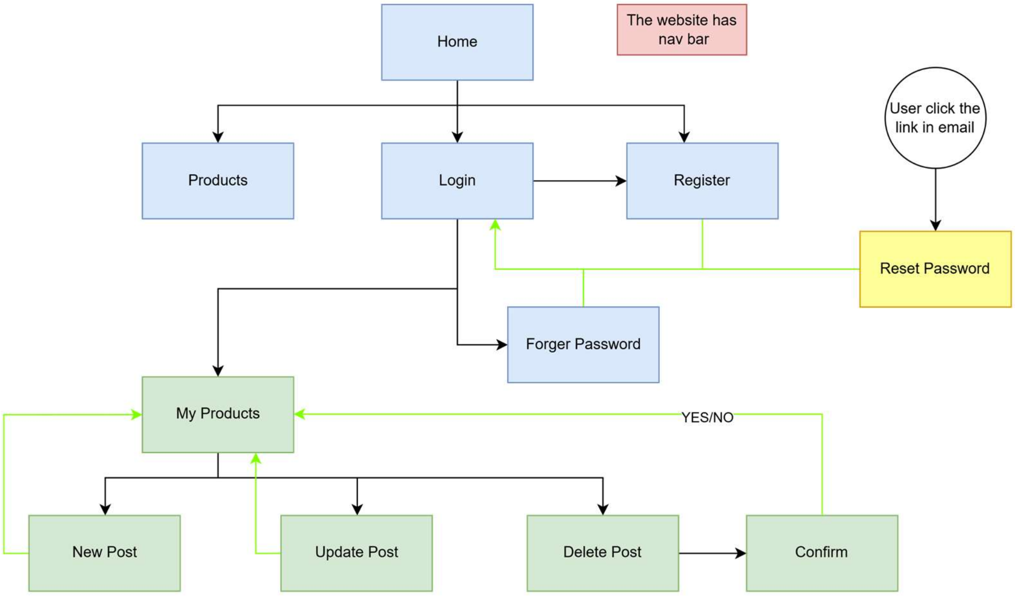
Flow Charts/Process Flows:

The flowchart below demonstrates the workflows of the website. As the user entered the URL, the home page will be open first, then using the navbar the user can navigate to others page. From the flowchart we can notify that the user can open login page, or the user can navigate to register page to register, also user can open the products page which the user can explore all the listed products. From the flowchart we notify that the user needs to register and login to access the futures of the website, such as create a post, update a post, or delete a post. Also, the flowchart shows the checks such as check for correct or valid username and password, as well as check validation of information that are entered by user when create or update a post. The PDF file for this flowchart is provided in Appendix B.


Start
User Enter URL
Home page
Products
Navbar?
Regester
Display Products
Register Form
Login
Forgot Password
NO
Enter valid email
Valid email?
YES
Email sent
Login Form
NO
Valid Username or password
Enter username and password
YES
User clicks the link in the email
Reset password form
Enter new password with confirm
Password met validation and do passwords matvh
NO
YES
Password updated
Create Post
My Products
Delete Post
Are you sure?
YES
NO
Product deleted
My Products
Create post form
NO
Valid Information
User enters product details
YES
My Products
Update Post
Update post form
NO
User updates product details
Valid Information
YES
My Products

Sitemap Diagram:

The sitemap diagram explains how the user will navigate through the website. As the flowcharts shows that the navigation will be through the navbar, in addition to the navbar some of my pages will have links to other page, for example, the login page will have a link to register page in addition to the navbar in case if the user has not registered. From the flowchart can be notified that the user after entering URL will be navigated to home page. Then the user will use the navbar to navigate through the website.



User Interface Diagrams:

For wireframe, please see Appendix B.

UML Diagrams:

For UML diagram, please see Appendix B.

Service API Design:

This project does not use any third-party API.

NFR’s (Security Design, etc.):

The non-functional requirement for this web application is security. While in this web application only authorized user can access the website features such as create a post, update, and delete, therefore, application will check for the user authentication, this will be checked when users enter their usernames and passwords to login. Also, the application encrypts user’s passwords by using BCrypt hash algorithm, which this will increase the application security. The other NFR is providing backup for database, which the database will be backed up every day after midnight, so the lost of data will be 24 hours.

Operational Support Design:

While the application is built using spring boot, the spring boot actuator will be used to monitor the application. The spring boot actuator module provides all of Spring Boot’s production-ready features. Also, actuator endpoints are other tools that will let the developer monitor the application. The Azure monitor is another tool that can be used to monitor the application. Azure monitor collects data from all the layers of application, azure monitor can check the application performance and identify the issue. Azure also provides Azure monitor log, which is centralized logging server that collect data from Azure and analyzing them, Log Analytics Workspace is used to store and query log. Spring boots provide logging tools such as Logback which is a logging framework that generates logs in applications that use java.

Other Documentation:

N/A.

Appendix A – Technical Issue and Risk Log

1. Use the template to identify and monitor project issues and risks.

Issues and Risk Log								
Issue or Risk	Description	Project Impact	Action Plan/Resolution	Owner	Importance	Date Entered	Date to Review	Date Resolved
Risk	Potential data breaches if passwords are not secured	User accounts could be compromised	Hash all passwords securely (BCrypt), use HTTPS for login/authentication	Soran Qiji	High	11/10/24		2/9/2025
Issue	Password reset	The users will not be able to use their accounts if they forget the password	The user will be able to change the password by requesting the link using email that is stored on database	Soran Qiji	High	11/10/24		3/24/2025
Risk	Image uploads	The user could upload other file types that could include Malware	Validating file type so that user can only upload image format	Soran Qiji	High	11/10/24		4/10/2025

Appendix B – References

File	Attachment
Wireframe.pdf	see the attached zip file
UML.pdf	see the attached zip file
Flowchart.pdf	see the attached zip file

Appendix C – External Resources

GIT URL:	*https://github.com/sqiji/capstone.git*