**CS673 Software Engineering** 

**Team 3 - ZicZac**

**Software Design Document**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Elijah Curme | Team Leader | *EC* |  |
| Dinara Tiyekbayeva | Configuration.Backup Leader | *DT* |  |
| Chenghao Feng | Security Leader | *CF* |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
|  |  |  |  |
|  |  |  |  |

[Introduction](#_87t9hln2vjz0)

[Software Architecture](#_buttcq9i221r)

[Design Patterns](#_x18fj36s1121)

[Key Algorithms](#_mtfbusfb0eq3)

[Classes and Methods](#_7ucksmkf6rzx)

[References](#_15tmymhipvdv)

[Glossary](#_8n34lvocupub)

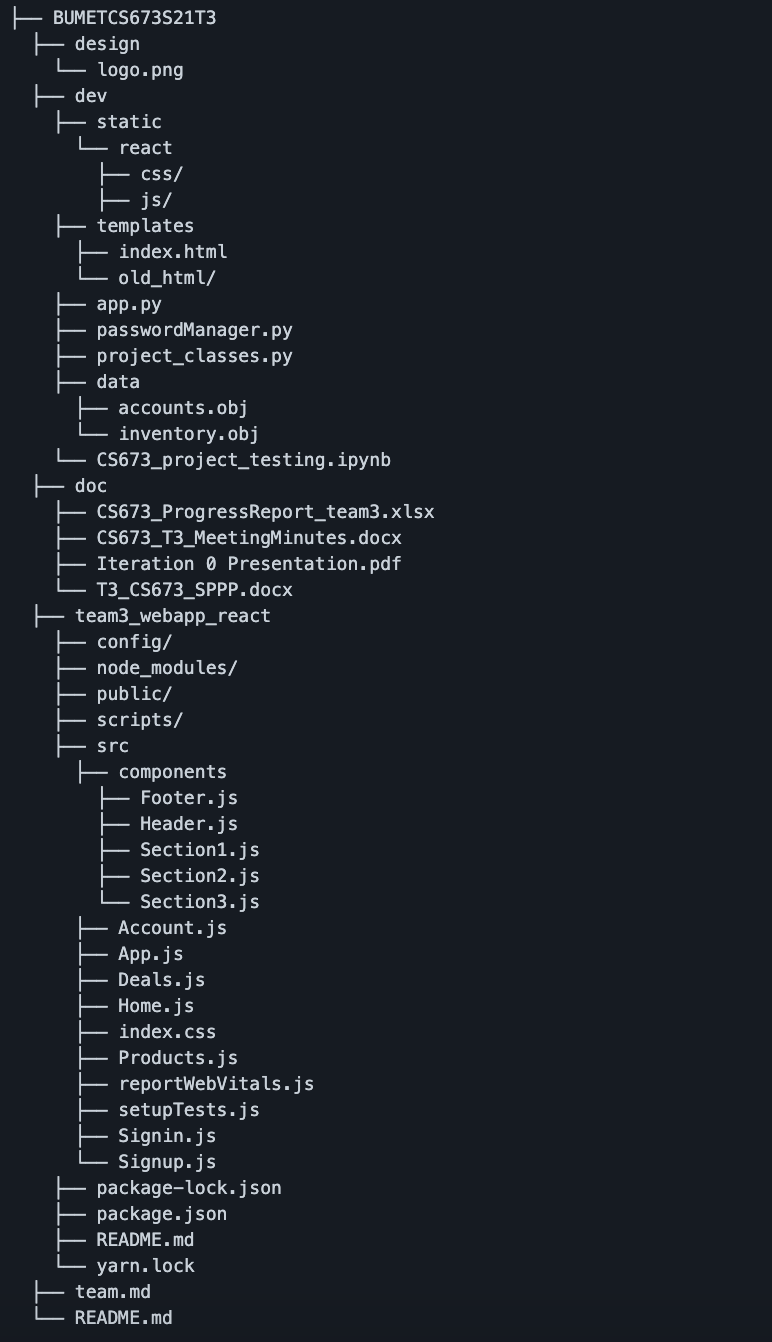
# Introduction

In this section, give an overview of this document, and also address the design goals of your software system.

# Software Architecture

In this section, you will describe the decomposition of your software system, which include each component (which may be in terms of package or folder) and the relationship between components. You shall have a diagram to show the whole architecture, and class diagrams for each component. The interface of each component and dependency between components should also be described. If any framework is used, it shall be defined here too. Database design should also be described if used.

Our software system is a React web application served by a Flask backend. The home page is composed of 3 primary section components and is bookended by header and footer components. Each React page includes a function that handles user input and returns html that is displayed by the browser. Backend tasks such as account verification or product filtering are performed by Flask, which cross-references application data in the data folder and returns results to React in JSON format. The project directory structure is shown in the below figure.



# Database Design (if applied)

In this section, you shall describe any database if used in your software system.

# Security Design

In this section, you shall describe any security design in your software system.

User’s account password follow the password standard such as minimum of eight upper- and lowercase alphanumeric characters, must include at least one special character (\*, &, $, #, !, or @). Additionally, SHA-3 and salt is used to encrypt the password while sign up and check while login in.

# Design Patterns

In this section, you shall describe any design patterns used in your software system.

# Key Algorithms

In our web application we are not inventing or manually writing algorithm solutions.

Instead of reinventing the wheel we are using ready-to-use solutions such as frameworks, external libraries, etc.

Product sorting:

Cryptography:

Chat system:

Search algorithm:

Load balancing:

# UI Design

Our goal was to make UI as easy as possible for users to interact with our application.

For designing User Interface we were using user stories from which we developed visual representations in justinmind wireframing tool.

Please see wireframing diagrams for complete understanding of UI used in this application.

# Classes and Methods

Please provide a link to your application API document which should be automatically generated.

# References

# Glossary