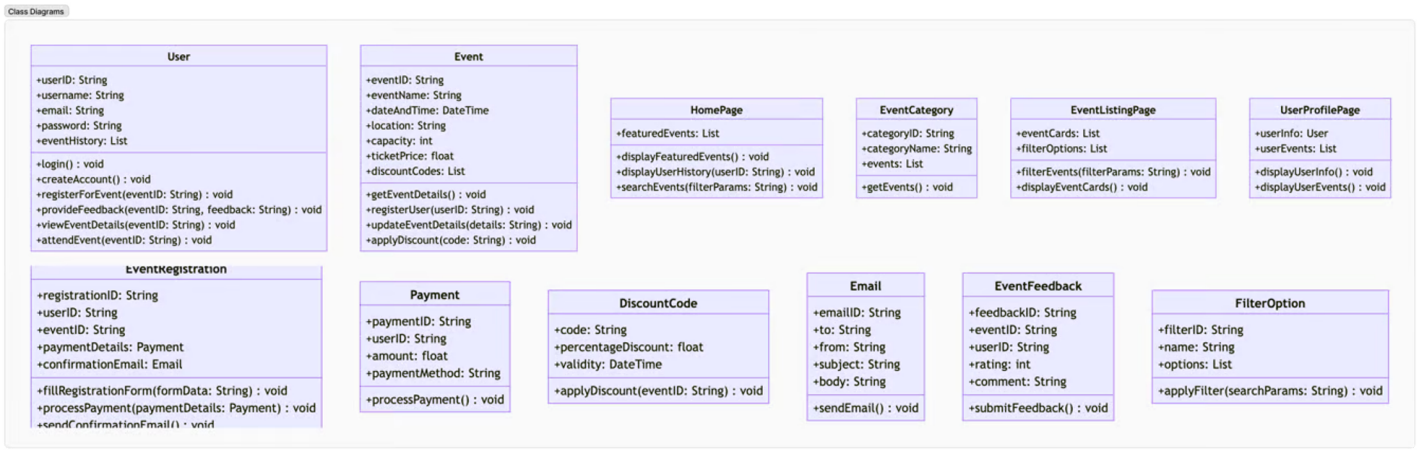
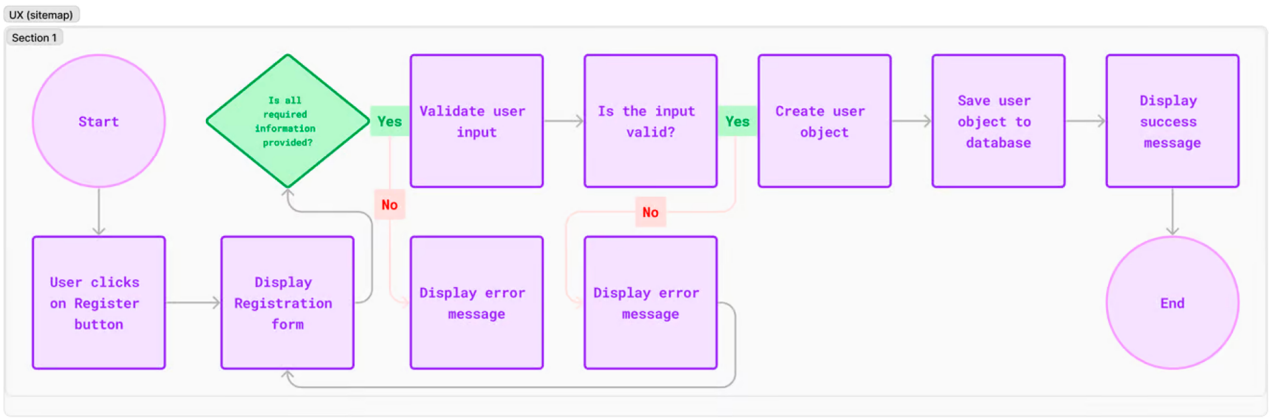
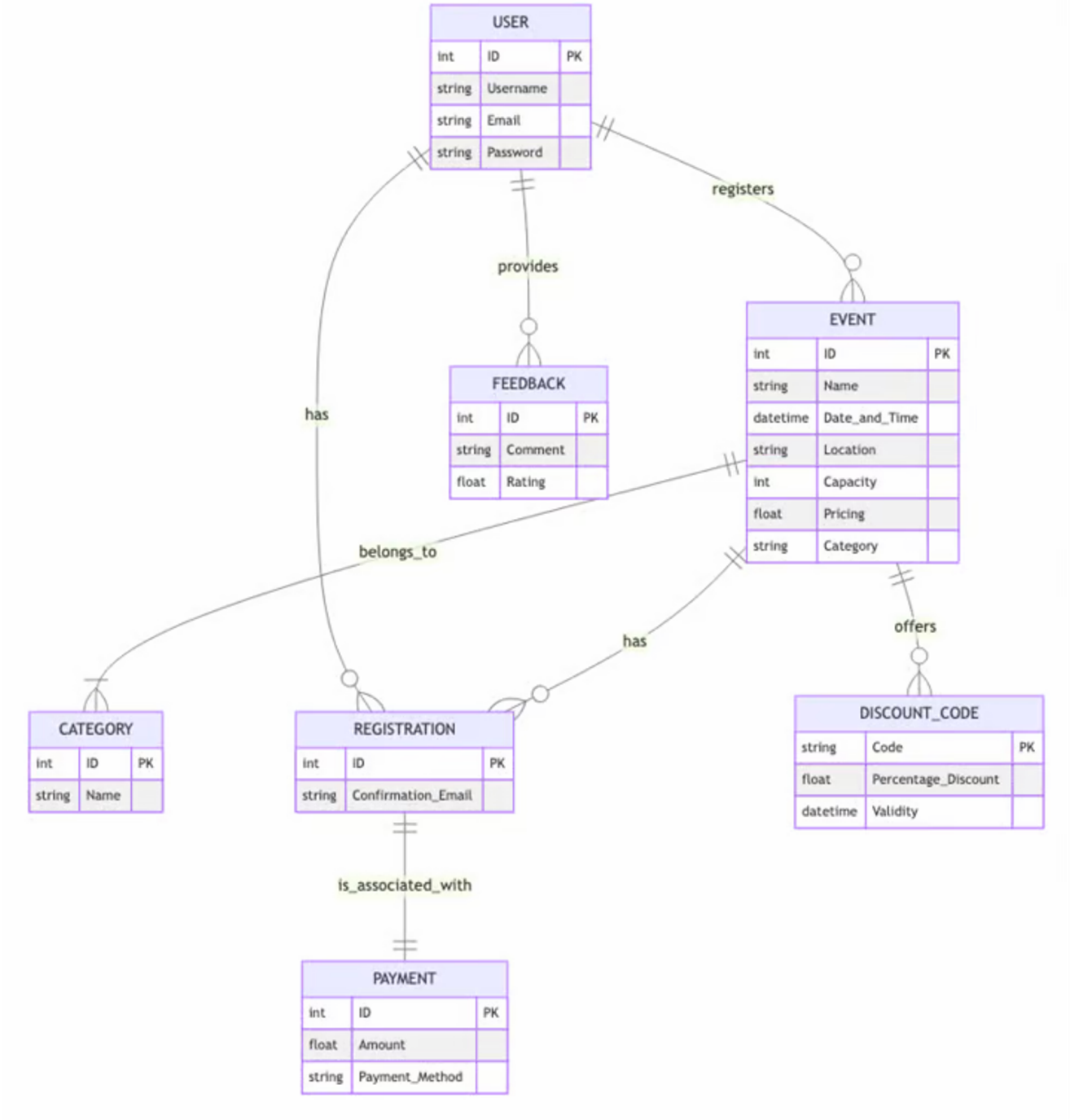
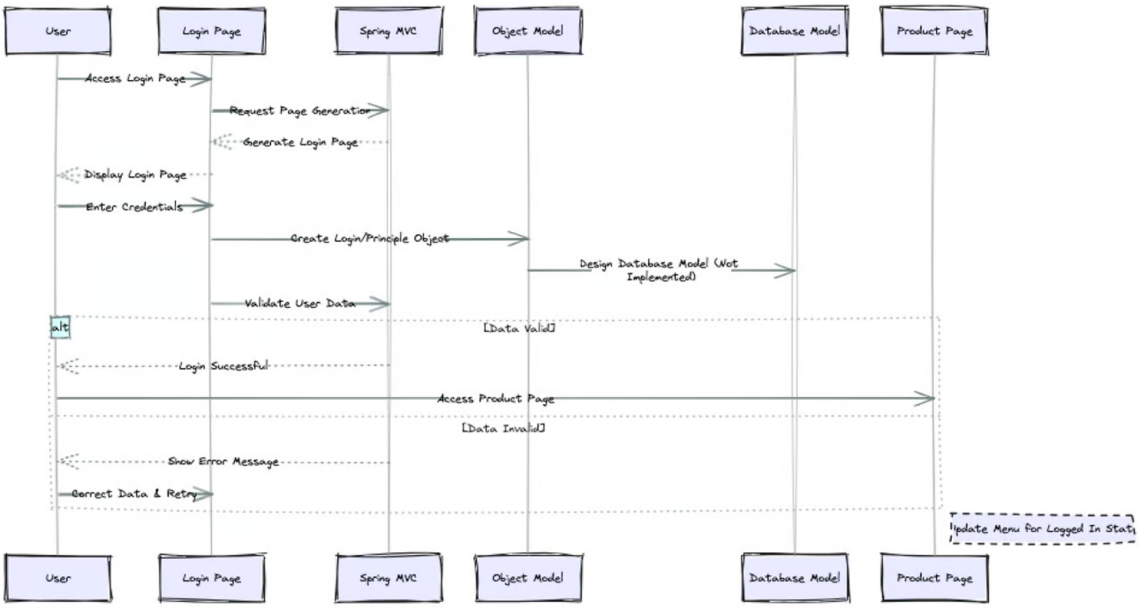
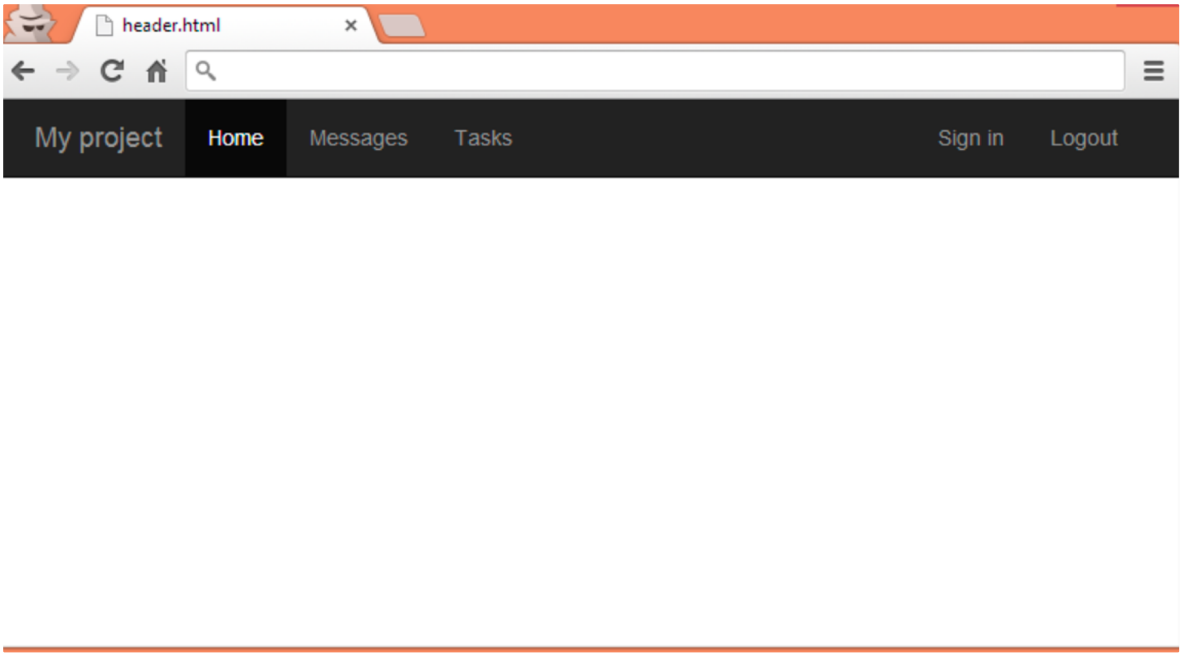
Milestone 2 Cover sheet
Alexander Johnson, Andrew Benavidez, Darnae Simmons
2 – 5 – 2024
CST-339
Diagrams and Completed Work






USER
int ID PK
string Username
string Email
string Password
registers
provides
has
FEEDBACK
int ID PK
string Comment
float Rating
EVENT
int ID PK
string Name
datetime Date_and_Time
string Location
int Capacity
float Pricing
string Category
belongs_to
has
offers
CATEGORY
int ID PK
string Name
REGISTRATION
int ID PK
string Confirmation_Email
DISCOUNT_CODE
string Code PK
float Percentage_Discount
datetime Validity
is_associated_with
PAYMENT
int ID PK
float Amount
string Payment_Method





Key Decisions
Like the previous week, we decided as a group to separate the work into three parts for us to work on to efficiently complete tasks. This week's
milestone requirements had three parts the group needed to work on so that made it easier to divide up the work evenly. Along with the coding
templates submitted in GitHub, the above screenshots are what the group completed this week.

UML:

Main Application Module

1. The main page appears when the root URI is accessed.
2. It provides initial navigation to access core functionalities like Login and Register.
3. It should have well-defined styles, fonts, colors, and an overall theme.
4. Spring MVC should be used for all dynamic page generation.

5. It should have a title (and possibly a logo).

Alex Johnson - Registration Module

1. Users should be able to register.
2. Registration requires First Name, Last Name, Email, Phone Number, and Login Credentials.
3. Spring MVC should be used for all dynamic page generation.
4. A user object model should be created.
5. A database model should be designed; implementation is for Milestone 4.
6. User data must be validated with clear error messages.

Darnae Simmons - Main Page

1. It shows up when you open the site.
2. It helps you get to key features like Login and Register.
3. It should look good with clear styles, fonts, and colors.
4. We'll use Spring MVC to make the pages.
5. It should have a title (and maybe a logo).

Sign Up

1. Users should be able to create an account.
2. To sign up, they need First Name, Last Name, Email, Phone Number, and Login Info.
3. We'll use Spring MVC to make the pages.
4. We need to make a user model.
5. We need to design a database model, but we'll build it in Milestone 4.
6. We must check the user data and show clear error messages if there's a problem.

Andrew Benavidez - Login Module

1. Initial login module should emulate user authentication.
2. Spring MVC should be used for all dynamic page generation.
3. A login/principle object model should be created.
4. A database model should be designed; implementation is for Milestone 4.
5. User data must be validated with clear error messages.
6. Security implementation using Spring Security is for Milestone 6.
7. After login, the main page should display and update to reflect the logged in state.

TBD - Sign In

1. The first sign-in feature should pretend to check user details.
2. We'll use Spring MVC to make the pages.
3. We need to make a login/principle model.
4. We need to design a database model, but we'll build it in Milestone 4.
5. We must check the user data and show clear error messages if there's a problem.
6. We'll add security with Spring Security in Milestone 6.
7. After signing in, the main page should change to show that the user is signed in.