CSCI 360 – Project Assignment Part 2

Due Date: 1 week from today (before class)

Submission Format: Single PDF or Word document per team (plus individual contribution summaries)

* **Michelle Clark Individual** P.A. Part 2 Assignment link: <https://docs.google.com/document/d/1KkgLBzRVAA41J-BxA1arESeNUI7YW5mk1Mfpkfm5cu0/edit?usp=sharing>
* **Joel Saldana Individual** P.A. Part 2 Assignment link: <https://docs.google.com/document/d/1aCtdxQ35ELcEhVAGwGRaruWELIZmr9-kj8Oq5L97Z1c/edit?usp=sharing>

**Team Assignment: System Architecture, UML Diagrams, and GitHub Integration**

This week’s assignment deepens your team’s design work and introduces essential tools used in modern software engineering. You’ll develop a visual system architecture, explore UML diagrams, and begin using GitHub for collaborative version control.

**Assignment Requirements**

1. **System Architecture Drawing** (20 pts)

Create a high-level architecture diagram that shows how your application will work. This should include:

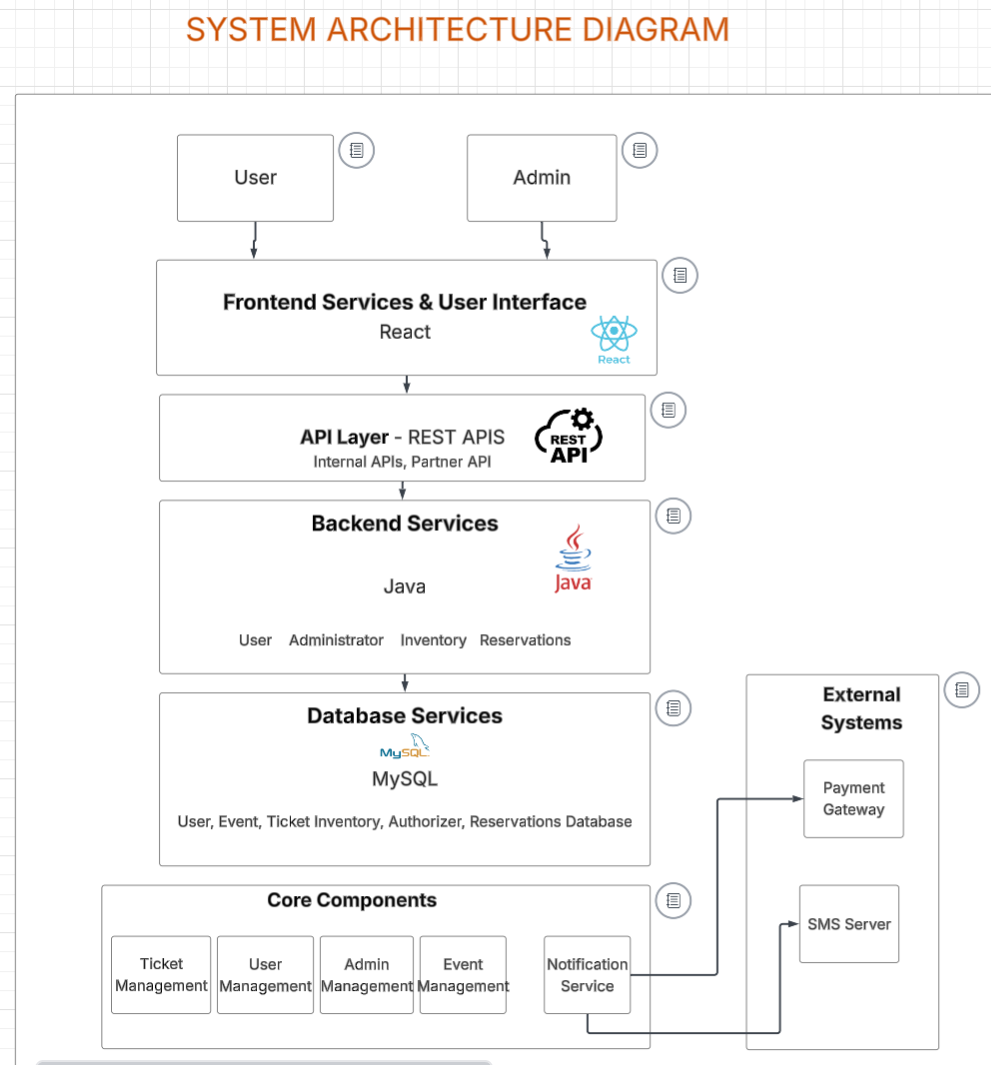
• Frontend technologies (e.g., React, HTML/CSS, etc.)

• Backend services (e.g., Node.js, API, database)

Any third-party APIs, authentication flows, or deployment services

• Diagrams can be made using Draw.io, Lucidchart, Figma, or similar

Include brief descriptions or annotations of each component.

* **Lucidchart sharable link:** <https://lucid.app/lucidchart/88d00304-aab8-4559-8642-c87cf33c34bb/edit?viewport_loc=-670%2C-700%2C3280%2C1738%2C0_0&invitationId=inv_f555984d-e0b7-40ff-9a97-723399400e8c>
* \*\*\*\* Annotations are made as notes in LucidChart (access by clicking the circle at the top right of the screen.
* << see annotations by clicking the circle in top right corner of each box in LucidChart

2. **UML and Sequence Diagrams** (20 pts)

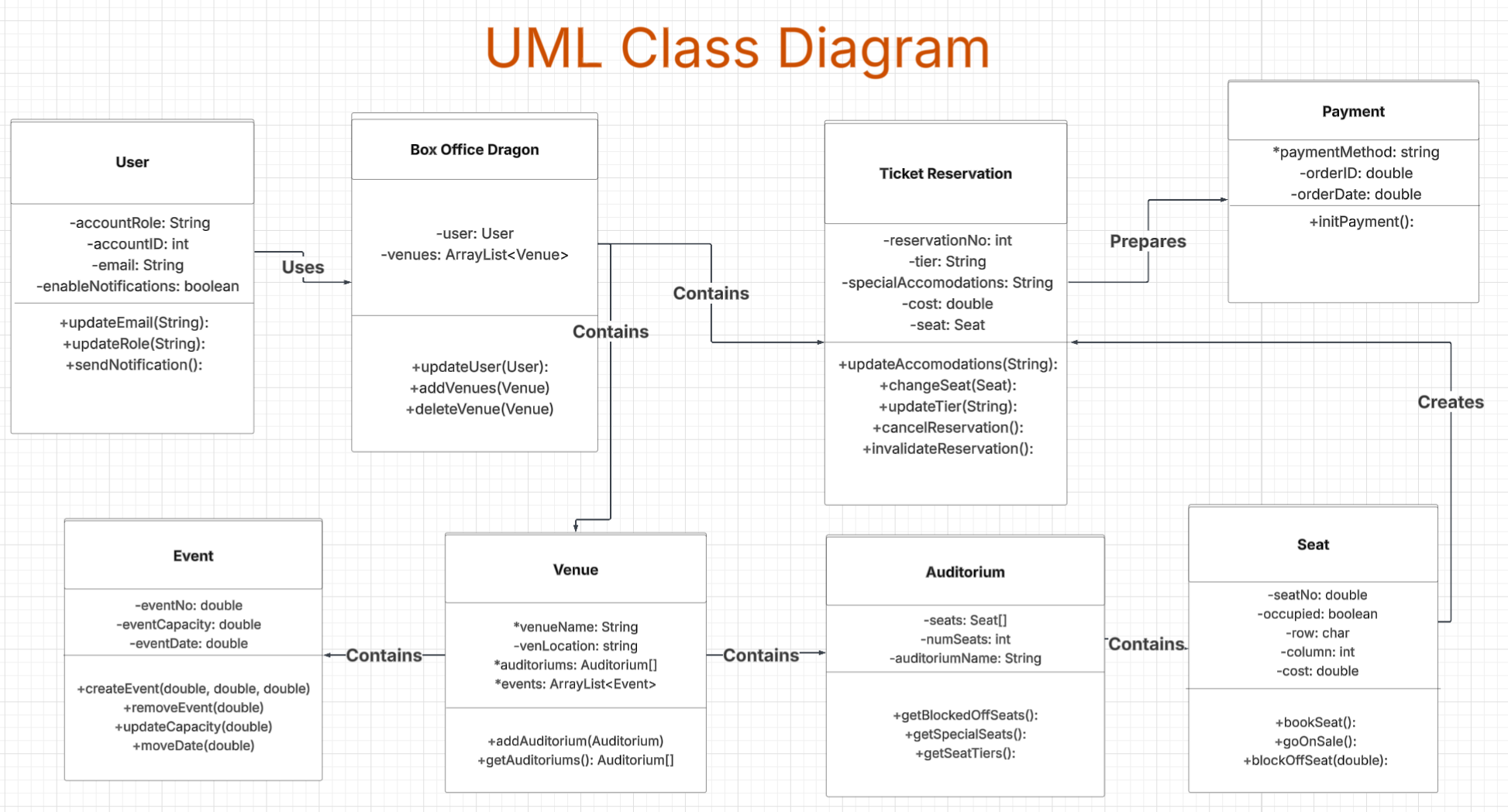
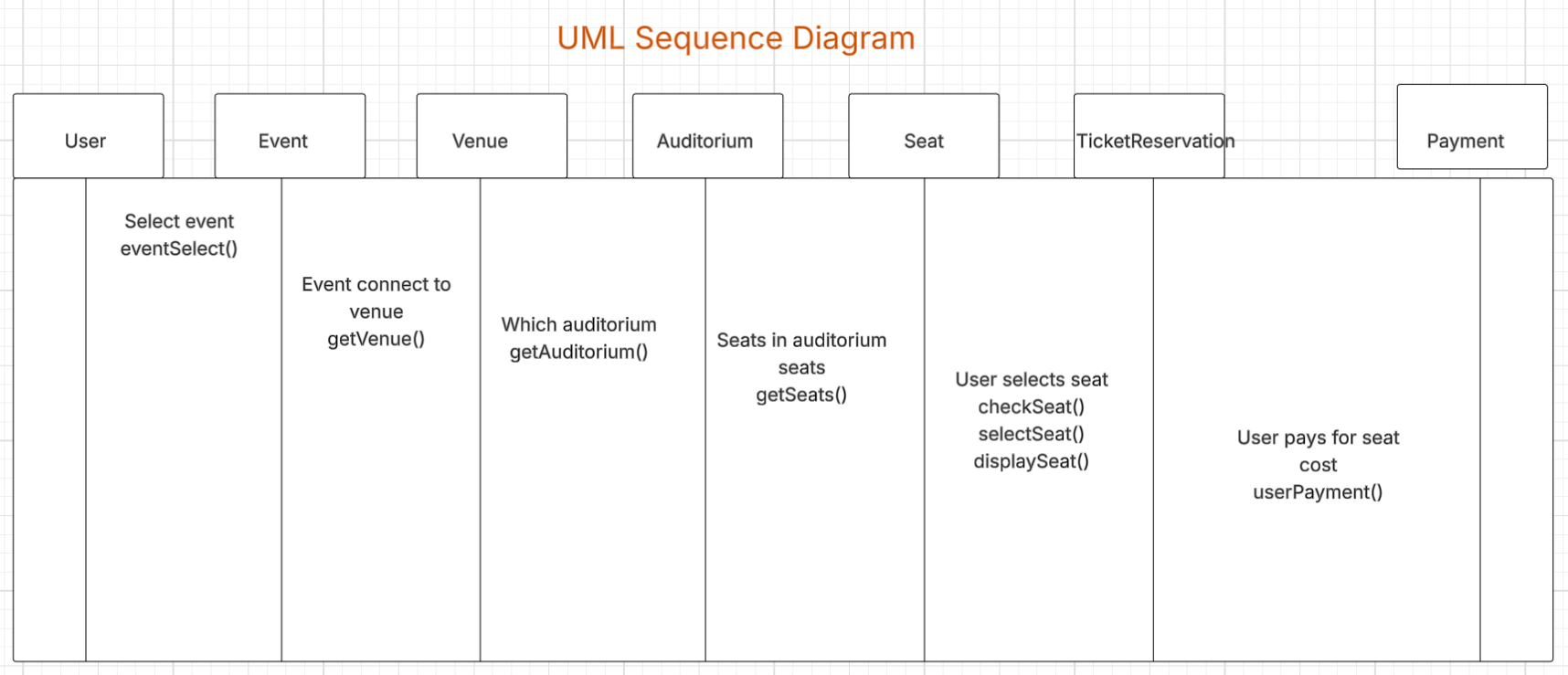
Create the following:

• One UML Class Diagram showing major classes/entities and their relationships

• One Sequence Diagram that illustrates a user purchasing a ticket (or similar key use case)

• Make sure diagrams are labeled clearly

• Tools such as Lucidchart, StarUML, PlantUML, or Draw.io are acceptable

* **Same Lucidchart diagram – each diagram labeled in red** <https://lucid.app/lucidchart/88d00304-aab8-4559-8642-c87cf33c34bb/edit?viewport_loc=-670%2C-700%2C3280%2C1738%2C0_0&invitationId=inv_f555984d-e0b7-40ff-9a97-723399400e8c>
* 
* 

3. **GitHub Repository Setup** (20 pts)

• Each team must create a GitHub repository for their project

• All group members must contribute at least one commit this week

• Add tony@tonymorelli.com as a collaborator on your private GitHub repo

• Provide the link to your GitHub repository in the submission

**Link to repo:** [**https://github.com/MM-Clark/CSCI360FinalProject.git**](https://github.com/MM-Clark/CSCI360FinalProject.git)

**4a. Individual Contribution - Summary (15 pts)**

Each student must include a clearly labeled section at the end of the submission with:

• A breakdown of who contributed to what (e.g., diagrams, architecture, code files)

• A list of GitHub commits made by each individual (copy/paste or screenshot accepted)

• Brief reflection (1 paragraph) on how the group is working together

Joel - UML Class Diagram lead

Vivian - UML Sequence Diagram lead

Michelle - System Architecture Drawing lead

All - slides, parts of each component

Reflection in individual contribution.

**4b. Individual Report - Design Patterns in Your Architecture (10 pts)**

This week we will also discuss common object-oriented design patterns in class. Each team must:

• Identify at least one design pattern that fits your project architecture

In individual contribution.

Strategy

• Justify why that pattern is appropriate for your use case

In individual contribution.

The pattern is appropriate for our use case because it allows us to define a family of payment algorithms and to select one at runtime. This allows us to support multiple third party payment systems in our app, as proposed in Assignment 1 (GooglePay, ApplePay, PayPal). We would have flexibility in our payment options, allowing us to simply create a new payment method through a new class that implements our payment strategy. Additionally, this pattern will help organize our checkout process and also aid in our concerns of sudden changes.

• Find and include one real-world example of a company or open-source project using that pattern (e.g., via engineering blogs, GitHub projects, technical case studies) This section will be graded based on thoughtfulness, clarity, and the connection to real-world usage.

In individual contribution.

**Ebay/Amazon**

* <https://medium.com/@elizgoodwin/how-amazon-uses-design-patterns-a2e17668d39b>
* Background: <https://refactoring.guru/design-patterns/strategy>

E-commerce websites such as Ebay and Amazon both utilize various payment methods. They use a microservices-based strategy pattern

**5. Presentation Prep (10 pts)**

Each team should be ready to give a 3–5 minute in-class update covering:

• Your architecture and diagrams

• GitHub setup and collaboration approach

• What you’re planning next

**Slides link:** <https://docs.google.com/presentation/d/1ODBJ9TKOalc1nW-AhZ_4t6FBIPu9-0Zem5_3Bs8UkGw/edit?usp=sharing>

**Submission Instructions**

• Submit a single PDF or Word document with all diagrams, repo link, and contribution summaries

• Each individual must submit a copy to OAKS with their own contribution summary included at the end

**Tips for Success**

• Don’t aim for perfection — your diagrams are expected to evolve

• Make at least one meaningful GitHub commit each: even small ones count

• Use version control best practices: meaningful commit messages, no broken code in `main`

• Keep your diagrams consistent with your use cases from Part 1