Teckit

Making finding, purchasing, and creating your own ticket easier than ever

Group Members: Demarcus Braclet, Aryan Kulkarni, Artan Vafaei, Sneha Gopalakrishan, Sakana Ali, Kai Castellanos, Kevin Dang

Objective of Teckit

- Develop an efficient software to sell, make, and purchase tickets
- Handle multiple users buying and refunding tickets simultaneously
- Handle site traffic spikes and prevent crash due to influx of users

Cost Effort Estimation

	Function Category	Count	Simple	Average	Complex	Count x Complexity
1	Number of user input	10	**3**	4	6	30
2	Number of user output	3	4	**5**	7	15
3	Number of user queries	6	3	**4**	6	24
4	Number of data files and relational tables	14	7	**10**	15	140
5	Number of external interfaces	0	5	**7**	10	0
					GFP	209

Reasonings for Cost

1. Login, Signup, Create Ticket, Edit Ticket, Delete/Refund Ticket, Create Event, Delete Event, Buy Ticket, Choose Seats,

2. Payment Info Receipt, Successfully Edit, Display Current Page

3. Lookup by Tickets, Lookup by Event, Lookup by artist, Look up Date & Time of event, Look up Location Seated Events, Event, Catalog, Seated Tickets, Ticket,

4. Account, Host Account, Admin Account, Buyer Account, Login, Database (data files, username, password, id's)

5. Not needed

Complexities: 4, 4, 5, 4, 3, 5, 5, 4, 2, 2, 1, 0, 0, 5

Computing Function Point

Compute gross function point (GFP).

$$GFP = (10*3) + (3*5) + (6*4) + (14*10) + (7 * 0) = 209 \text{ FP}$$

Determine processing complexity (PC).

$$E = FP / \text{productivity} = 209 / 60 = 3.48 \approx 4 \text{ person - weeks}$$

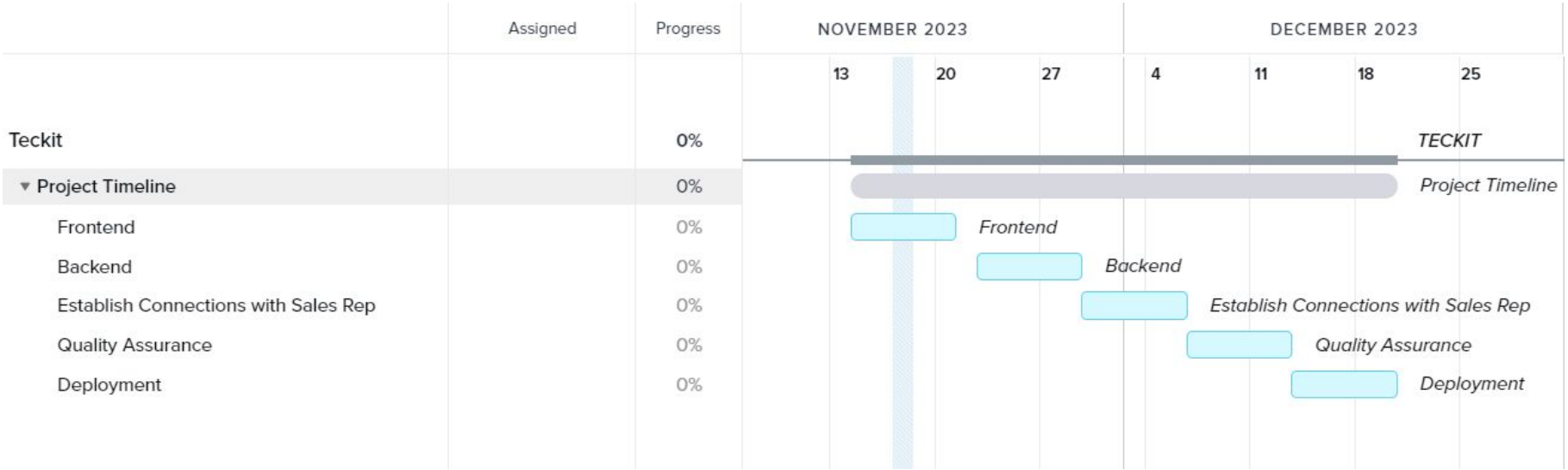
Compute processing complexity adjustment (PCA).

$$PCA = 0.65 + 0.01 \text{ x } ((4 * 4) + (4 * 5) + (2 * 0) + (1 * 3) + (1 * 1) + (2 * 2)) = 0.65 + 0.01 \text{ x } 44 = 1.09$$

Compute function point (FP) using the formula: FP = GFP × PCA

$$FP = 209* 1.09 = 227.81 \text{ FP}$$

Project Timeline



Functional Requirements

1. User Authentication
    a. Allow users to create and login using their own credentials on the website.
2. Ticket Display and Search
    a. System must display trending tickets on homepage, system must provide categories for user to search by, users should be able to search for products using search bar.
3. Secure Checkout
    a. The system should ensure secure payment when processing with credit cards and other methods such as PayPal
4. Data Encryption:
    a. All user login information and data collected must be encrypted using latest encryption standards. In addition, the website must have an SSL certificate to ensure secure connections
5. Shopping Cart:
    a. The system must provide a shopping cart with a checkout feature. Users should be able to add items to their cart, delete items from their cart, and save items for later.

Non-functional Requirements (Product)

1. User interface should adhere to WCAG to ensure accessibility for all users

2. Efficiency Requirements:
    a. Performance: Website should load pages and process transactions under 2 seconds for 95% of user interactions
    b. Space: Data center hosting system should have sufficient space for current and future servers

3. Dependability Requirements: System should have a maximum scheduled downtime of 4 hours per month for maintenance

4. Security Requirements: All data must be encrypted with industry-standard encryption algorithms.

Non-Functional Requirements (Organizational)

1. Environmental Requirements: Data center and server infrastructure should be designed with energy efficiency in mind

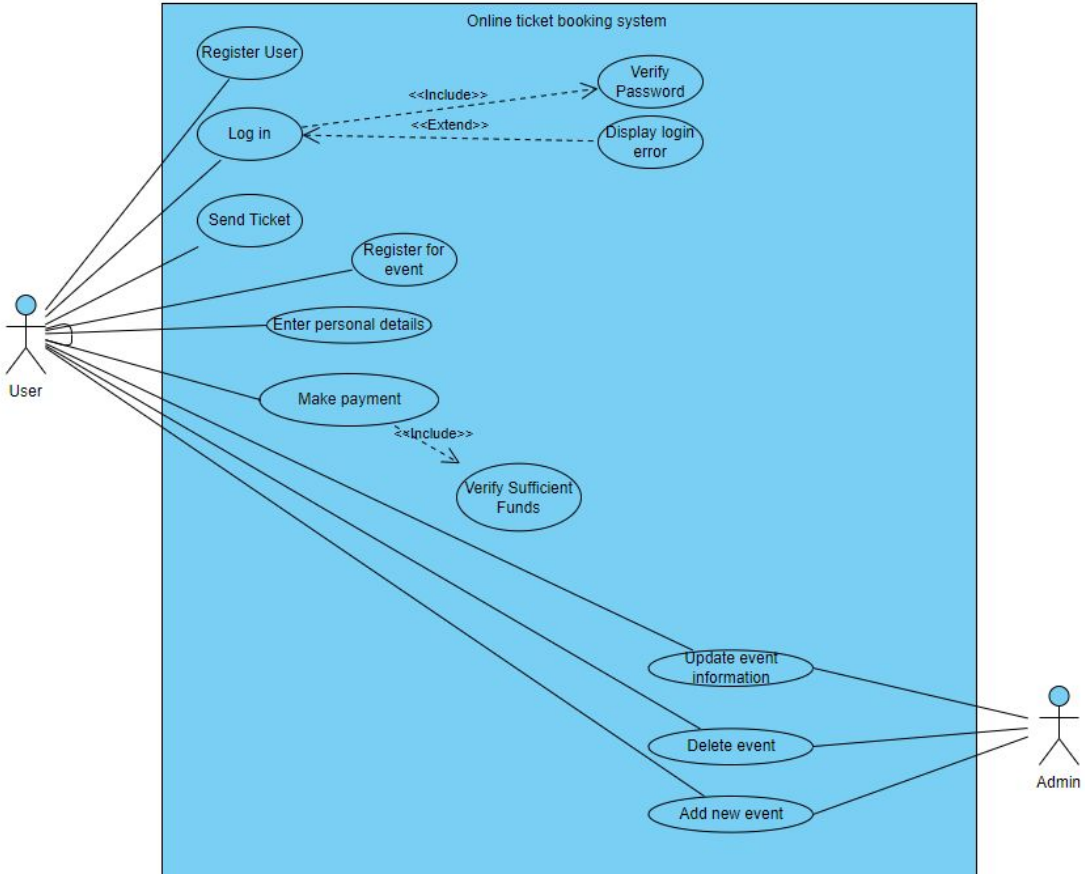
2. Operational Requirements: The system should be available 24/7 with a maximum of 0.1% unscheduled downtime per year

3. Development Requirements: Development team will follow agile methodology with bi-weekly sprints.

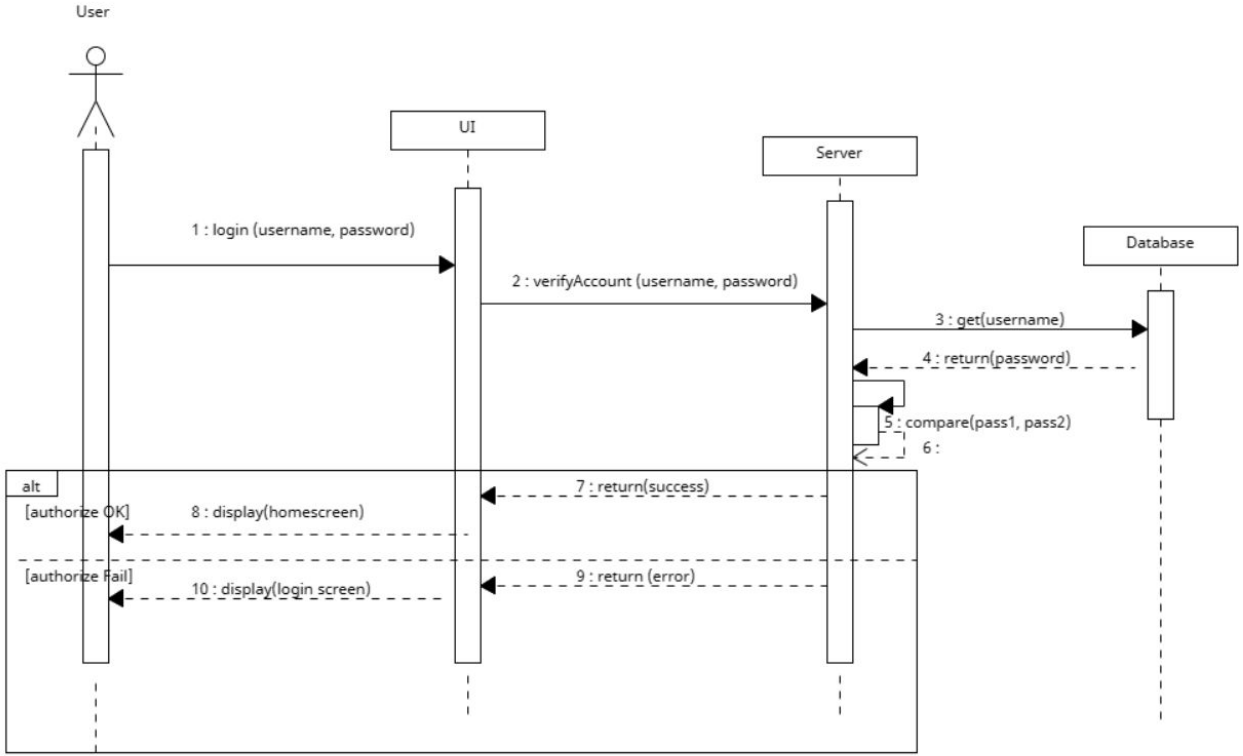
Non-Functional Requirements (External)

1. Regulatory Requirements: The system must comply with CCPA for user data privacy and consent.
2. Ethical Requirements: Content moderation will be in place to prevent promotion of harmful content on the platform.
3. Legislative Requirements: Assuming a product release in the US, the organization must comply with US legislative requirements.
4. Accounting Requirements: The system will maintain detailed financial transaction logs for auditing and accounting purposes
5. Safety/Security Requirements: Access to production servers will be restricted to authorized personnel and a robust access control system will be in place to maintain security.

Use Case Diagram



Sequence Diagrams (Log in)



Tekcit Login

Email or Phone

Password

Forgot password?

Login

Not a member? Signup now

TEKCIT

Find Ticket Pricing FAQ Blog Create Ticket

Quickly Sell, Buy And Create Your Very Own Ticket

Teckit Making finding, purchasing, and creating your own ticket easier than ever

GET STARTED NOW →

Events | Parking | VIP | Others

No events within 50 miles of New Hope for all dates

91 events in all locations

NOV 17
FRI
7:00 PM
Today

Taylor Swift Tribute
Teatro Ramon Lopez Velarde
Fenaza, Mexico

See Tickets

NOV 17
FRI
7:30 PM
Today

Taylor Swift
Estadio Nilton Santos-Engenhao
Rio de Janeiro, Brazil

See Tickets

NOV 17
FRI
8:00 PM
Today

Taylor Swift Laser Dance Party
Mr. Smalls Theatre
Millvale, PA, USA

See Tickets

NOV 17
FRI
9:00 PM

Taylor Swift Night - The Taylor Party
Rams Head Live
Baltimore, MD, USA

See Tickets

Venues

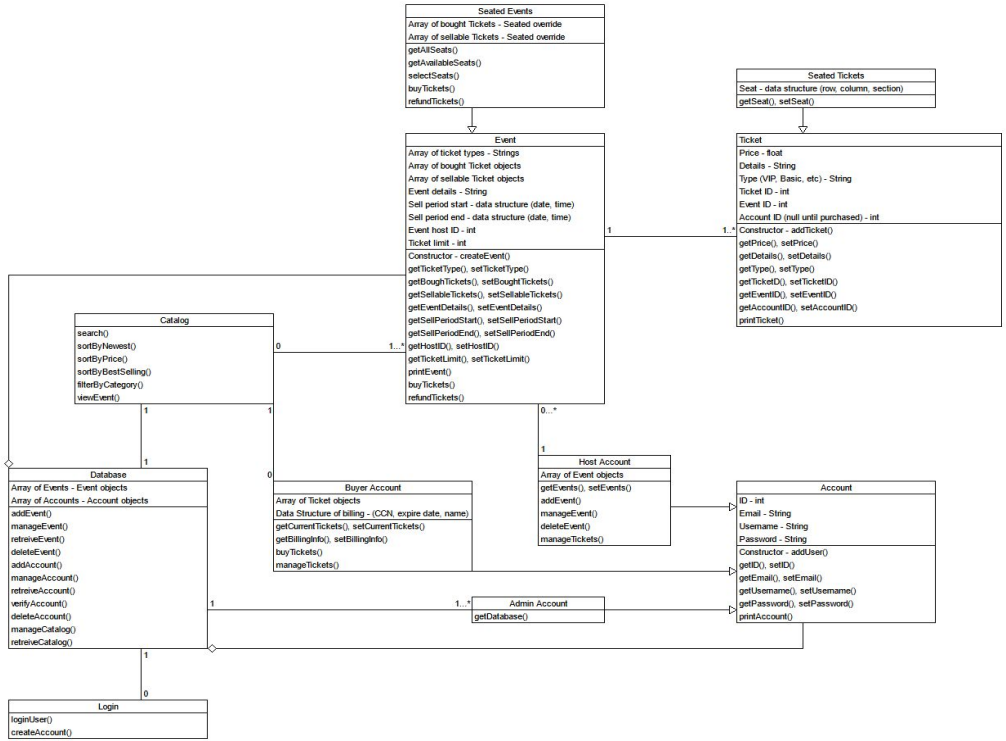
Singapore National Stadium
Singapore, SG

Anfield Stadium
Liverpool, GB

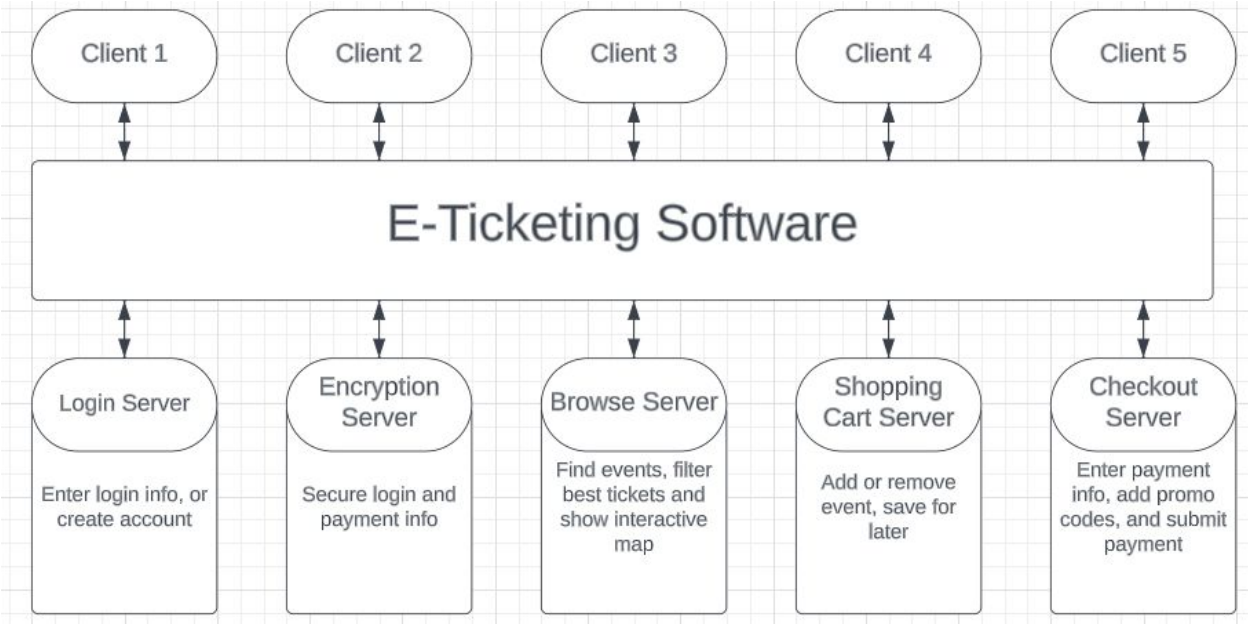
Estadio Nilton Santos-Engenhao
Rio de Janeiro, BR

See More

Class Diagram



Architectural Design (Client-Server)



Comparison with Ticketmaster

Teckit

- Scalability is designed to handle large influx of concurrent users
- Reduces traffic and potential server overloads
- Robust support staff equipped with necessary tools for customer assistance
- Implements anti-scalping measures

Ticketmaster

- Large pre-sales and extreme surge in users results in website failure
- Does not have anti-scalping measures
- Customer Support lacking
- Speed suffers during high demand

Conclusion

- Designed software for Teckit by incorporating and optimizing aspects of pre-existing software such as TicketMaster.
- Utilized client-server architectural design system to reduce complexity.
- Sequence and Use Case diagrams created to demonstrate how the software would run.
- No major changes from initial design due to thorough research.
- Following the Software Engineering design process greatly assisted in reducing the complexity of designing the software.

Thank you!

Any questions?