**Milestone 2**

Bree Hardy

Jacob Harris

Aleney Gonzalez-Martinez

Department of Computer Info Systems CIS, Davenport University

CISP 111

Professor Krogmann

November 21, 2024

**Milestone 2**

This paper will go through the process of the planning phase for creating a mobile app named *The Movie Vault,* to turn a movie theater to every fan’s dream of new releases and classical films. The main objectives are to set a clear plan with a timeline, budget, and any possible challenges, so the app meets user needs and is practical to build. This process includes looking at how people buy tickets now, providing clear guidance for developers, and making sure the app is easy to use, efficient, and works well. Testing will confirm that it performs as expected and meets all objectives before it's released to users. The focus is on understanding how people will interact with the app, what data it needs, and ensuring the design matches user expectations.

**Phase 1: System Planning**

The planning phase for developing the mobile app’s main feature, buying movie tickets, is the first step. This phase helps us decide what the app should do, like letting users browse current and classic movies, choose seats, buy tickets, and get deals on food. It involves figuring out what users want and what features are most important, such as seeing available seats in real time and offering safe payment options. As the creators, it is important to see if the project is possible, looking at time, money, and resources needed. The goal is to make a plan with a budget, timeline, and any possible challenges, ensuring the app will be useful and meet users' needs.

The purpose of *The Movie Vault* is to provide users with an easy way to purchase movie tickets, view classic and current movie listings, select seats, and access special deals on food; all through an intuitive mobile application.

**Primary objectives:**

* Enable users to easily purchase tickets for movies in two locations.
* Provide an easy interface for viewing seating charts and selecting seats.
* List both classic and current movies with showtimes.
* Showcase deals on food combos to enhance the user experience.
* Ensures secure and smooth payment processes.

**Key features to meet the user needs includes:**

* Movie Listings: Show current and classic movies with detailed descriptions, ratings, trailers, and showtimes.
* Seat Selection: Visible seat maps showing availability and seat prices.
* Multiple Locations: Allow users to switch between two different cinema locations.
* Deals and Offers: Provide users with combo deals on food, displayed during the booking process.
* Payments: Support multiple payment gateways and ensure secure transactions.

Additionally, the users will be able to have an opportunity to create an account that will store their past movie bookings, preferences, and most importantly payment information for easy payment for future transactions. With the creation of an account, they will also be allowed to receive notifications, if they choose. If enabled they will be notified about movie deals, upcoming shows, and booking reminders.

**SWOT Analysis**

Strengths: Its easy-to-use interface, which lets users browse current and classic movies quickly, helps movie fans find their favorite films. It also has useful features like real-time seat selection and secure payment options that improve user satisfaction and encourage more ticket purchases. Additionally, offering exclusive deals on food can attract more users and give the app an advantage over competitors.

Weaknesses: Integrating real-time seat availability and ensuring a smooth payment process could be technically challenging and take a lot of time. The app’s success depends on forming partnerships with movie theaters, and if these partnerships don’t work out, it could limit how many users the app reaches. While keeping the app updated and running smoothly it will require ongoing maintenance, which can increase long-term costs.

Opportunities: The app can reach a wide audience by targeting moviegoers who want a simple way to buy tickets, especially those interested in both new releases and classic films. Forming partnerships with theaters could lead to exclusive promotions, boosting downloads, and getting more people engaged. Additionally, adding features like virtual movie screenings or loyalty programs could help keep users coming back and attract new customers.

Threats: It competes with well-known ticketing platforms that offer similar features, making it important to stand out in the market. Technical problems including slow loading times or payment issues, could frustrate users and lead to negative reviews. Lastly, changes in user habits, such as more people preferring to stream movies at home rather than going to theaters, could lower the demand for buying movie tickets in person.

**System Requirements for a successful opertable application**

Hardware:

* Mobile devices like iOS and Android that will support the app.
* Backend servers for hosting movie databases, user data, and handling transactions.

Software Requirements:

* Mobile development frameworks like Flutter. Backend databases like MySQL for storing movie, location, and user data.
* Integration with payment gateways like paypal.

Network Requirements:

* Reliable internet access for users to interact with the app.

The last system requirement is an application programming interface. This will allow a connection between the app and movie theater systems for up-to-date information on seat availability and deals.

**Roles and Responsibilities to build *The Movie Vault*:**

* Project Manager: Oversees the overall project, timeline, and delivery.
* Developers: Mobile app developers for iOS and Android (or cross-platform), backend developers for server-side functionality.
* Designers: UI/UX designers to create an intuitive and visually appealing interface.
* Quality Assurance Testers: Ensure the app is free from bugs and functions as intended.
* Marketing Team: For promoting the app once it is ready for release.

**Collaboration Plan:**

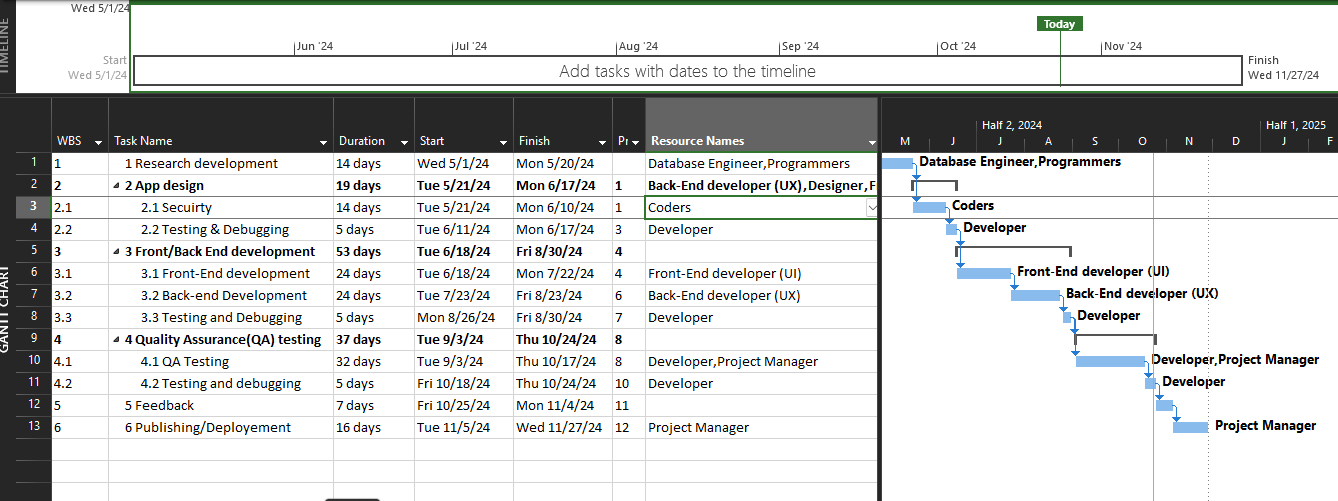
According to Tilley (2019), “An agile approach emphasizes continuous feedback, and each incremental step is affected by what was learned in the prior steps” (p. 1.7.3). Our team will utilize agile methodologies, incorporating regular communication practices such as daily stand-up meetings, sprint planning sessions, and continuous integration. The use of project management tools, Jira, will aid in tracking progress and maintaining project transparency.

**Costs Estimates:**

The estimated costs for the applications development and maintenance

* Outsourcing costs for developers and designers. ($50/hr)
* UI/UX design, including wireframes and prototypes. ($46/hr)
* Server costs, database management, and API integrations. ($60/hr)
* Budget for marketing the app before and after launch. ($5,000)
* Ongoing costs for bug fixes, updates, and customer support of project management tools like Jira to track progress. ($2,500-$4,000)

**Estimated Schedule/Time Frame:**

**Figure 1: Gantt Chart**

The major activities of the Planning Phase focus on building an efficient foundation for developing *The Movie Vault*. These activities include identifying the app’s key features, like real-time seat selection and secure payment options, and understanding what users need to have a smooth experience. It also involves analyzing the feasibility of the project by considering the time, budget, and resources required. Creating a detailed plan with a clear budget, timeline, and potential risks ensures that the development process is structured and goal-oriented. Overall, this phase is a need for setting clear objectives and a solid foundation, guiding the project towards success.

**Phase 2: System Analysis**

The Analysis Phase is about understanding the details of what the app needs to do. This includes gathering information through observations and looking at how people currently buy tickets. It also involves creating diagrams to show how information, like movie listings and payments, will flow through the app. The goal is to understand how users will use the app and what data needs to be stored and processed. This phase includes making a model to show how data is stored and how data moves to ensure the app works smoothly. It also looks at using methods like Agile to help the team work together better. The goal is to make sure the app's design meets all user needs and works well.

**Requirements of Engineering**

*What does this mean?* Requirements Engineering involves gathering, specifying, validating, and managing the requirements of the system to ensure the final product meets user needs (Jedrzejko, 2023). It is essential to categorize these requirements into:

**Functional Requirements to identify the functions the system must perform.**

* User account creation and login.
* Browse current and classic movies.
* Selection of showtimes and locations.
* Seating map interaction (view, select seats).
* Payment integration for purchasing tickets and food.
* Generate and manage e-tickets.

**Non-Functional Requirements to describe system attributes like performance and security.**

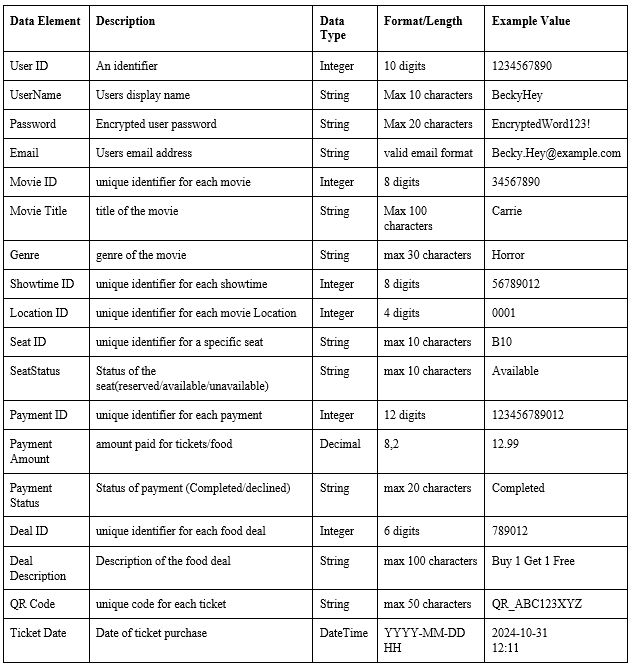
* App responsiveness (2-second load times for movie listings)
* 99.9% uptime availability. (always up to date and operable for users)
* Agreement with data protection standards
* Scalability for future cinema locations and payment methods.
* Compatibility with Android and iOS platforms.

**Fact-Finding Techniques**

Our team will *observe* how users currently purchase tickets at the cinema to identify potential pain points that could be lessened by the app. Gathering this information, we will use Excel to organize and analyze survey data, calculate system performance, and document requirements. This will track project timelines, milestones, and task progress with Gantt charts in Excel.

**Data and Processing Modeling**

**Data Dictionary:** Provides a clear reference for creating tables, fields, and relationships. By defining each data element explicitly, our team can avoid misunderstandings and reduce errors during development and integration. It helps in maintaining consistency and provides a shared understanding of the data structure throughout the development process.

**Figure 2: Physical Data Model Structure**

**Users Table:** Stores user information for account management and login.

* UserID (Primary Key): Unique identifier for each user.
* UserName: User's display name.
* Email: User’s email address.
* Password: Encrypted user password.
* PhoneNumber: Contact number for account security and notifications.
* DateCreated: Timestamp of account creation.

**Movies Table:** Stores information about each movie.

* MovieID (Primary Key): Unique identifier for each movie.
* MovieTitle: Title of the movie.
* Genre: Movie genre (e.g., Action, Comedy).
* ReleaseDate: Date the movie was released.
* Rating: Movie rating (e.g., PG-13).
* Description: Brief description of the movie.

**Showtimes Table:** Holds data on when and where each movie is playing.

* ShowtimeID (Primary Key): Unique identifier for each showtime.
* MovieID (Foreign Key): Links to Movies table to specify which movie is playing.
* LocationID (Foreign Key): Links to Locations table, indicating the cinema.
* ShowDateTime: Date and time of the show.
* ScreenNumber: Number of the screen where the movie is shown.

**Locations Table:** Stores information about the cinema locations.

* LocationID (Primary Key): Unique identifier for each location.
* LocationName: Name of the cinema.
* Address: Physical address of the cinema.
* PhoneNumber: Contact number for customer service.

**Seats Table:** Stores seat information for each showtime.

* SeatID (Primary Key): Unique identifier for each seat.
* ShowtimeID (Foreign Key): Links to the specific showtime in the Showtimes table.
* SeatNumber: Number of the seat (e.g., A10, B5).
* Status: Indicates if the seat is *Available* or *Reserved.*

**Payments Table:** Manages payment information for transactions.

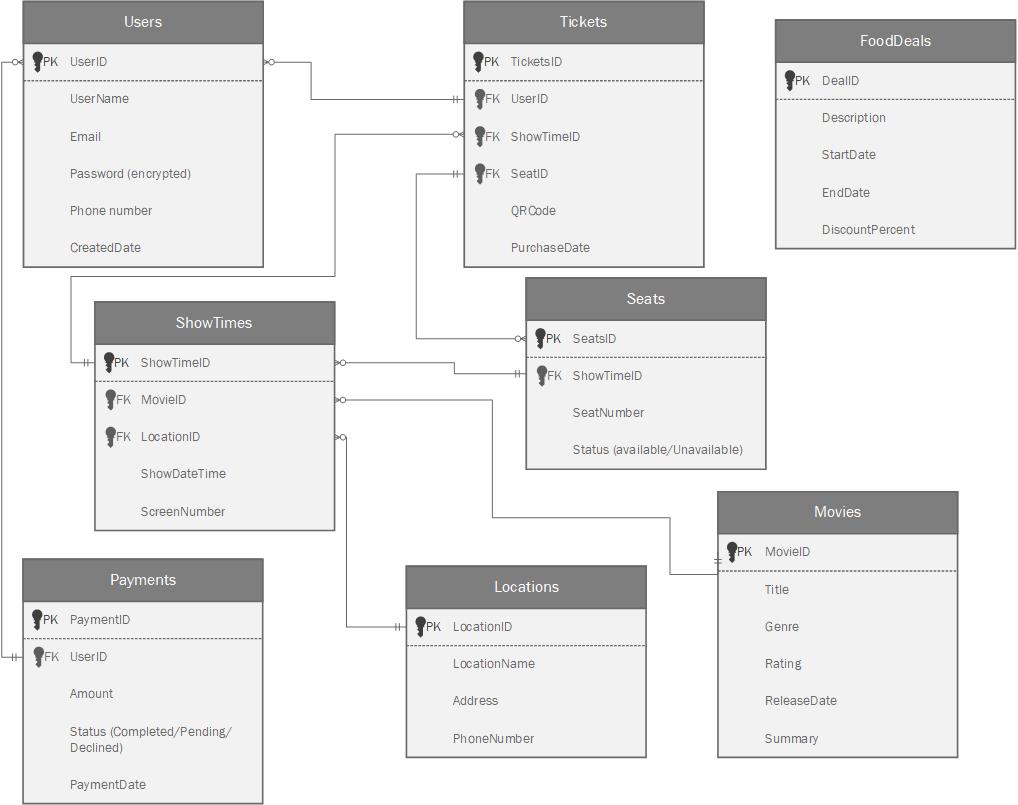
* PaymentID (Primary Key): Unique identifier for each payment transaction.
* UserID (Foreign Key): Links to the Users table for identifying the user making the payment.
* Amount: Amount paid by the user.
* PaymentStatus: Status of the payment (e.g., Completed, Pending).
* Timestamp: Date and time when the payment was made.

**Tickets Table:** Tracks individual ticket purchases and generates a unique ticket per user.

* TicketID (Primary Key): Unique identifier for each ticket.
* UserID (Foreign Key): Links to the Users table.
* ShowtimeID (Foreign Key): Links to the Showtimes table.
* SeatID (Foreign Key): Links to the Seats table.
* QRCode: Unique QR code for ticket validation at entry.
* PurchaseDate: Date and time of ticket purchase.

**Phase 3: System Design**

In the Design Phase of the app, we created plans for the app’s user interface and how it will work behind the scenes. Key screens include the Home Screen with movie listings, the Movie Details Screen for showtimes, the Seat Selection Screen with an interactive seating chart, and the Checkout Screen for payment options. Each screen is set up to be easy to use and allows quick access to food deals and account settings. We designed a database to organize key information, like users, movies, showtimes, and payments, so the app can check seat availability and handle orders. Diagrams show how data moves through the app, especially during ticket booking and payment. This design ensures the app will be secure, reliable, and enjoyable for users.

**Figure 3: Database Schema Design**

**API and Integration**

“API integration is what opens a channel that enables our companies to, quite literally, conduct business faster and more accurately” (Keeports, 2024). The app will need third-party integrations for secure payments or if possible movie metadata. The API endpoint is a specific location within an API that accepts requests and sends back responses.

Payment Gateway Integration:

* Purpose: Process user payments securely.
* Example: PayPal API
* Accepts the request and send responses (End Points):
* /create-payment-intent: Generates payment details.
* /confirm-payment: Verifies payment status and returns confirmation.

Movie Metadata API (Optional):

* Purpose: Pull metadata on upcoming movies.
* Example: IMDb.
* Accepts the request and send responses (End Points):
* /search-movies: Fetch movie details by title.
* /movie-details: Get specific details like cast, genre, or trailer links.

Email/SMS Notification API:

* Purpose: Send confirmations and reminders.
* Example: Twilio for SMS, SendGrid for emails.
* Accepts the request and send responses (End Points):
* /send-email: Email purchase confirmation to the user.
* /send-sms: Send ticket QR codes or reminders.

**Testing and Security Plan**

Plan:

* Functional Testing: Validate that each feature works correctly (e.g., movie browsing, seat selection, food deals).
* Performance Testing: Test load handling during peak times (e.g., popular movie releases).
* Integration Testing: Check all APIs (payment, notifications) function correctly within the app.
* Security Testing: Test for vulnerabilities, including secure login, payment encryption, and data access controls.

Security Measures:

* Data Encryption: Encrypt all sensitive data (passwords, payment details) with AES-256 encryption.
* SSL/TLS Encryption: Use HTTPS for all data transmissions.
* PCI-DSS Compliance: Ensure payment processing aligns with PCI standards for data protection.
* Access Control: Limit data access within the app and implement role-based permissions for administrative function

**Phase 4: System Implementation**

The main focus of the implementation phase is ensuring that the app is easy to use for not only the customers but the employees as well. This will require “...constant attention to quality assurance throughout the activities of application development, coding, testing, documentation, and installation” (Tilley, 2019). For quality assurance, a user evaluation form will be sent to the user’s email address and if filled out they will earn an incentive that could range from a free small popcorn to a discount on their next purchase.

**Training guide for employees:**

Objective one: Purchase movie tickets

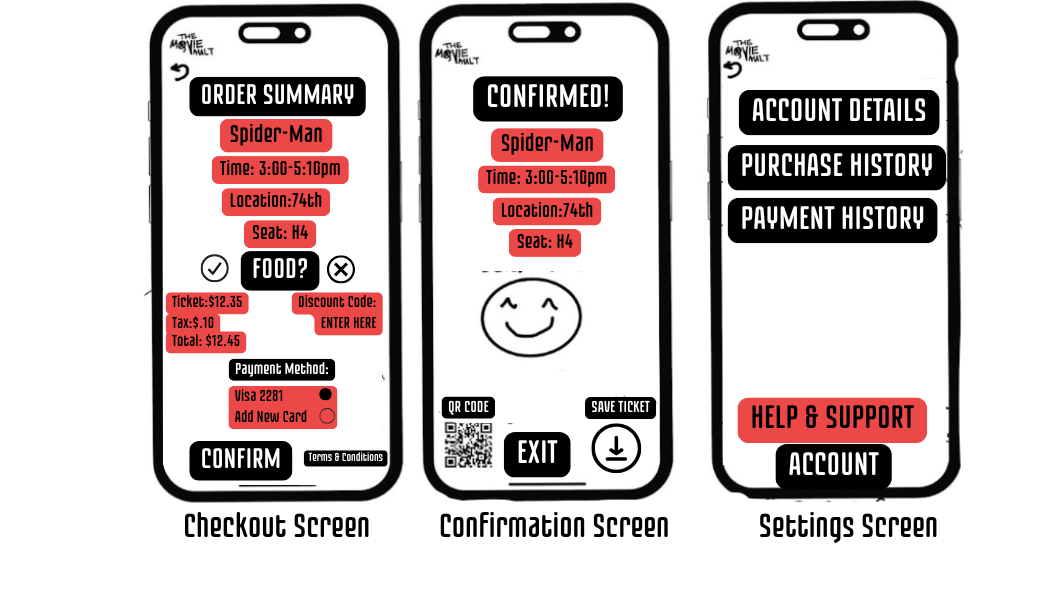
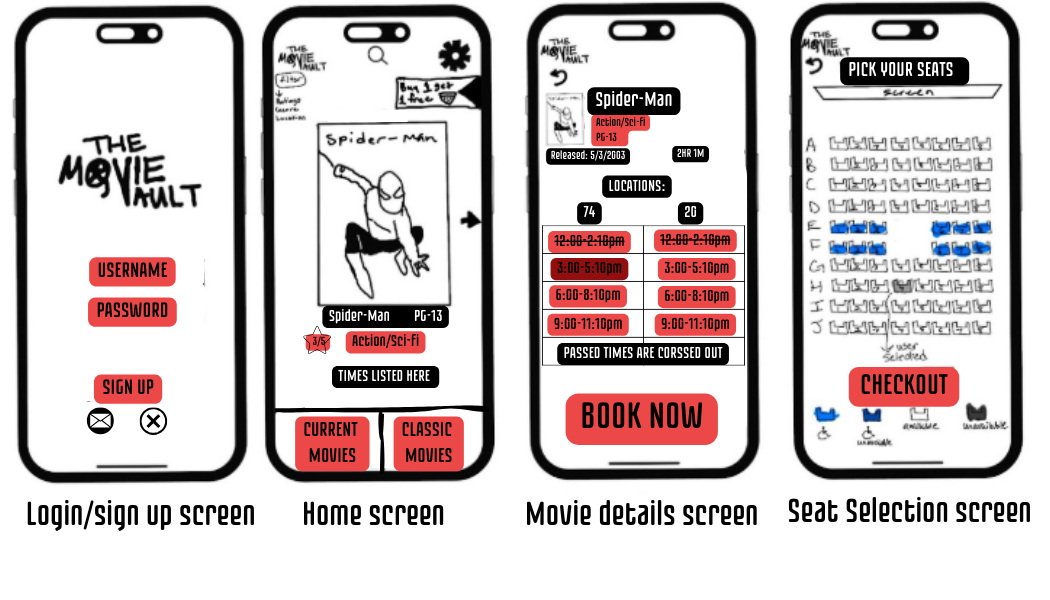
Figure 4 demonstrates how the app looks in terms of the user. It would prompt the customer to either make an account or to continue as a guest. Now it will take the user to the home screen, and notice that it will alert the returning customers of their rewards in the top right corner. Along with showcasing a featured movie that week could be a classic or a current one. Now onto the movie detail screen, select a location and time if the time slot is sold out it will be labeled as *Sold Out* and if the time slot has passed it will be crossed out/grayed out. Seat selection screen, on the bottom of the screen it has a color-coded key. The checkout screen will provide an order summary of what the user is about to pay for. The last prompt will be asking if the user would like to add on food. If selected no user will put in a payment method and be sent straight to the confirmation screen. If selected, yes it will bring users to a new screen and the ability to see what that specific theater offers in terms of food and sizes with prices. Lastly the confirmation screen will confirm everything that was bought and selected with a QR code and the ability to save the ticket to their camera roll.

Objective two: Refund a ticket or food concession

With the QR code that is presented in Figure 4 scan it into the computer. With the QR code, the employee can see the tickets bought, time, concessions, and location. The employee can change movie screening, the movie selected, the location, and most importantly a refund button. The employee will begin selecting the items the customer would like to refund and once completed an email will be sent to the customer and a refund receipt will print.

Objective three: Navigating the user account and settings

By clicking settings it will reveal account details, purchase history, payment settings, and online help and support. In the account details, the customer will find their personal information such as full name, email, username, phone number, and how long the customer has been an account holder. In the purchase history, the customer can find all their past purchases through the app. In the payment tab, the customer will see the cards that have been saved on their account and the ability to remove, add, or edit the payment details. Lastly is the help and support section, this is where the customer can submit a question and they are guaranteed a response in 24 hours.

**Figure 4: The Movie Vault App Design** 

**User Evaluation:**

The goal of a user evaluation is to ask simple questions that are designed to achieve a goal which in this case is to improve the app for the user (Hajesmaeel-Gohari et al., 2022).

* On a scale of 1-5, 1 being the worst and 5 being the easiest. How easy is the usability of the app?
* What was your favorite feature of the app?
* Would you prefer ordering your tickets online or the more traditional way of going in person to order the movie tickets?
* Did you encounter any difficulties while using the app?
* What is one thing would you change to make the app more user-friendly?

**Phase 5: System Support and Security**

This final phase focuses on user support, known as customer service, the help part of the application for users to use if they need help or have any problems with the app. After this development, there is a maintenance and management process where the software is monitored, updated, and fixed based on user feedback and changing requirements. Accordingly to Ahmed, “Security is now widely recognized as crucial to a successful SDLC, and incorporating security operations across the SDLC aids in the development of more dependable software.” (Ahmed, 2024) Another focus is on system performance management, monitoring the application for signs of trouble fault management, and performance and workload measurement. Also the system security and the levels of security of the database and application. Furthermore, with support and security, there is also systems backup and recovery for any application failure, and creating a business continuity plan (BCP) for how our application can continue during any major disruption. Lastly the system retirement and the future challenges and opportunities of technology and our application.

System Support (Customer Service, maintenance task and management, and system performance management):

* Login Issues: Forgot password providing a link, account not found double checking you are typing in the correct information, and account lockout
* Payment Problems: Payment not going through, and incorrect billing information
* Ticket Booking/Buying Issues: Unable to buy tickets check to make sure still in stock at available location, seats not showing try refreshing, error on checkout try to refresh if the problem continues to delete and redownload.
* App Crashes or Freezes: If the app is not responding, try restarting or checking for updates, or persistent issues, go into your device's settings and clear the app's cache (for Android) or reinstall the app.
* Notifications not working: Not receiving alerts check to make sure push notifications are located in settings, and email notifications either subscribe for them or unsubscribe to not receive.
* Refunds or Cancellations: Requesting a Refund by checking our policy and seeing if a refund is available in the time range, canceling a reservation If the app allows it, go to your *purchase history* section to cancel your reservation. If you're unable to do so, contact customer support for assistance.
* Technical Support: Contacting customer support look for a *Help & Support* option in the app's menu, to chat with someone for help, app feedback if facing bugs, or having issues contact and email bugs or contact customer support
* Location and Showtimes: Showtimes not displaying make sure the right time and location are set, and the active location is turned on.

System Security follows our previous security measure put in place in phase 3. Data Encryption, SSL/TLS encryption, PCI-DSS compliance, and Access Control.

System Retirement: Make sure to ensure a smooth transition for all stakeholders and customers.

* Purpose of Retirement: Transition to a new platform, obsolescence due to outdated technology, maintenance challenges or high costs, decline in user adoption, or business strategy shift.
* Stakeholder Communication: notify all stakeholders customers, employees, and partners
* Retirement Timeline: Establish key milestones, announcement, data migration period, feature phase-out, final deactivation, and post-retirement support
* Data Management: user data handling and backend data archival
* Feature Phase-Out: Disable key features in ticket booking, app updates, and account creation.
* Alternative Solutions: Redirect to new or alternative programs, or redirect to third-party partners offering partnerships.
* Technical Shutdown: System decommissioning, removing the app from app stores (Google Play, App Store), disabling backend servers, APIs, integrations, and ensuring that all sensitive data is securely deleted from servers.
* Legal and Financial Considerations: Review contracts, User refunds, and liability management.
* Customer Support and Feedback: Support channels for customers through the retirement period, and feedback collection for the next program.
* Post-Retirement Activities: Review success metrics, redirect resources, and documentation of retirement.

**References**

Ahmed, A. (2024, June 28). A detailed discussion on SDLC 5 phases in 2024 - Hapy CO. A Product Building Agency.<https://hapy.co/journal/sdlc-5-phases/>

Hajesmaeel-Gohari, S., Khordastan, F., Fatehi, F., Samzadeh, H., & Bahaadinbeigy, K. (2022). The most used questionnaires for evaluating satisfaction, usability, acceptance, and quality outcomes of mobile health. *BMC Medical Informatics and Decision Making*, *22*(1). <https://doi.org/10.1186/s12911-022-01764-2>

Jędrzejko, K. (2023, July 6). The importance of software requirements engineering in IT projects. *Software Mind*. <https://softwaremind.com/blog/software-requirements-engineering-the-driving-force-behind-successful-and-efficient-it-projects/>

Keeports, A. (2024). What is API integration?. Cleo. [https://www.cleo.com/blog/what-is-api-integration#:~:text=API%20integration%20refers%20to%20the,to%20communicate%20with%20each%20other.](https://www.cleo.com/blog/what-is-api-integration#:~:text=API%20integration%20refers%20to%20the,to%20communicate%20with%20each%20other)

