Fully-Dressed Use Cases

Use Case Section	*Comment*
Use Case Name	Purchasing a ticket
Scope	Java-lagged team designed event ticketing software
Level	User Goal
Primary Actor	User
Stakeholders and Interests	- User: wants to purchase a ticket - College of Charleston: wants to receive money from the transaction and ensure an accurate inventory of tickets
Preconditions	- The ticketing software is running and executing normally - The user has found whatever play being searched for and proceeded to checkout - There exists some ticket in the inventory that is able to be purchased
Success Guarantee	- Transaction is recorded and performed accurately - The ticket is provided to the user without issue - Receipt is also created and distributed to the reader - Ticket inventory is updated accordingly
Main Success Scenario	1. The user enters in their payment information 2. The user confirms the purchase

	3. The transaction completes successfully and the ticket is digitally sent to the user 4. The system updates the ticket inventory to account for the purchased ticket
Extensions	2a. The user attempts to purchase a ticket but there are no tickets available 1. A warning will be displayed to the User that no tickets remain for this event 2. They will be notified of any similar events on other days that have available tickets 3a. The transaction fails to complete 1. An error will be displayed to the user and an admin will be alerted to the failure 2. The ticket that was attempted to be purchased will be held in reserve for 24 hours or until an admin can resolve the error and allow the user to purchase the desired ticket 3b. The ticket is never sent 1. An error message will be sent to both the User and an admin 2. The system will allow the user to choose between receiving a full refund or waiting until an admin is able to resolve the issue and send them a ticket 4a. The ticket inventory is not updated/updated incorrectly 1. The system will attempt to check

	for discrepancies in ticket inventory and alert an admin 2. An admin with appropriate authority will enter the database manually and correct the ticket amounts 3. The system will be reviewed to figure out what caused the discrepancy and how to fix it
Special Requirements	- Java - Compatibility with different payment methods
Technology and Data Variations List	3a. Payment system that can interface with different payment providers and securely complete transactions 4a. Database that can store ticket information
Frequency of Occurrence	Almost continuously
Miscellaneous	- Should users be allowed to store payment information on their account so that it can be autofilled in the checkout screen?

Use Case Section	*Comment*
Use Case Name	Log in
Scope	Java-lagged team designed event ticketing software
Level	User Goal
Primary Actor	User

Stakeholders and Interests	- User: wants to log into their account
Preconditions	- The ticketing software is running and executing normally - The user already possesses a valid account
Success Guarantee	- The User enters their login information accurately - The system successfully validates the login information and grants the user access to their account
Main Success Scenario	1. User navigates to the login button 2. On the login screen, the user enters their username and password 3. The system checks the username and password and recognizes them as valid 4. The user is granted access to their account
Extensions	2a. The user attempts to login with incorrect login information 1. An error is reported to the user 2. The user is allowed to try and input their login information up to 5 times 3. If 5 attempts are exceeded, the user will be directed to reset their login information 3a. The system fails to recognize the username and password as valid (even though they are) 1. An error is reported to the user

	2. The user will be prompted to re-enter their login information 3. If the error persists, the user will be directed to reset their login information or submit a bug report to the system 4a. The user is not granted access to their account 1. The system will log the error and direct the user to try and login again 2. Repeated failed attempts will signal the System to contact the Administrators about the error
Special Requirements	- Java
Technology and Data Variations List	2a. A database that can be used to store and verify login information
Frequency of Occurrence	Almost continuously
Miscellaneous	- Will login need to use the same two-factor authentication as creating an account?

Use Case Section	*Comment*
Use Case Name	Creating an account
Scope	Java-lagged team designed event ticketing software
Level	User Goal
Primary Actor	User
Stakeholders and Interests	- User: wants to create an account

	to use this software - College of Charleston: wants to track the number of people using this software
Preconditions	- The ticketing software is running and executing normally - There is space in the database to store the login information of another account
Success Guarantee	- A new account is successfully created
Main Success Scenario	1. User navigates to the "Create Account" button 2. User enters a valid username and password that has at least one Uppercase, one lowercase, a number, special character, and a minimum of 10 characters 3. The user will be prompted to verify their identity through a 3rd-party authenticator 4. The system creates a new account and automatically logs the user into it
Extensions	2a. The user enters an invalid username or password 1. The user will be informed that their username/password is invalid and prompted to enter a new one 3a. Two-factor authentication fails to validate the identity of the user 1. The system will display an error to the user 2. The user will be prompted to

	reattempt two-factor authentification 4a. The system fails to create a new account 1. An error will be reported to the user and to the System Administrators 2. The user will be directed to try creating another account
Special Requirements	- Java - Two-Factor Authentication
Technology and Data Variations List	2a. A password verifier that will ensure all password meet the minimum password requirements 3a. A 3rd-party authenticator app that can be used for two-factor authentication 4a. Database that can store account information for users
Frequency of Occurrence	Almost continuously
Miscellaneous	- What application should be used to facilitate two-factor authentication?

Use Case Section	*Comment*
Use Case Name	Search for a play
Scope	Java-lagged team designed event ticketing software
Level	User Goal

Primary Actor	User
Stakeholders and Interests	- User: wants to find a specific play - College of Charleston: wants to sell tickets for that specific play
Preconditions	- The ticketing software is running and executing normally - There exists some play in the System that the user wants to find
Success Guarantee	- The user's desired play is successfully found
Main Success Scenario	1. The user navigates to the search bar 2. The user searches for the desired play using some criteria (i.e. name, date, etc.) 3. The system returns the correct play that the user desired
Extensions	3a. The system fails to return the correct play that the user desired 1. The system will report to the user that 0 results were found for their query 2. The user will be prompted to attempt a new query or else search for a new play entirely
Special Requirements	- Java
Technology and Data Variations List	1a. A search bar where the user can enter in some criteria and have the database return all plays matching that criteria
Frequency of Occurrence	Almost continuously

Miscellaneous	- How should the returned plays be sorted and displayed to the user? - How many plays should be displayed to the user at once?

Use Case Section	*Comment*
Use Case Name	Sorting Plays
Scope	Java-lagged team designed event ticketing software
Level	User Goal
Primary Actor	User
Stakeholders and Interests	- User: wants to sort existing plays based on some criteria (i.e. alphabetically, chronologically, etc.)
Preconditions	- The ticketing software is running and executing normally - Existing plays are able to be sorted
Success Guarantee	- Existing plays are sorted based on User criteria - The sorted plays are displayed to the user
Main Success Scenario	1. The user selects “Advanced Filter” 2. The user chooses an option to sort by (Alphabetical, Chronological, within a certain period of time) 3. The plays are sorted and

	displayed to the user
Extensions	3a. The plays are sorted incorrectly 1. The user will have the ability to submit a bug report indicating that the sorting feature is incorrect
Special Requirements	- Java - A Sorting Algorithm
Technology and Data Variations List	2a. A sorting algorithm that can efficiently and accurately sort the events in the database and then display them to the user
Frequency of Occurrence	Almost continuously
Miscellaneous	- What specific sorting algorithm should be used? - Would different sorting algorithms be needed based on what criteria the user wishes to sort by?

Use Case Section	*Comment*
Use Case Name	Selecting Seats
Scope	Java-lagged team designed event ticketing software
Level	User Goal
Primary Actor	User
Stakeholders and Interests	- User: wants to select a specific seat or seats for their chosen play
Preconditions	- The ticketing software is running and executing normally

	- There exists some seats available to be chosen by the user - The user has already selected a play
Success Guarantee	- The system accurately records the seat(s) chosen by the user
Main Success Scenario	1. The user navigates to the seat selection screen before they purchase their ticket 2. The user selects a vacant seat or seats that they wish to reserve 3. The user confirms their selection and proceeds to checkout
Extensions	2a. The user selects a non-vacant seat 1. The user will be notified that the seat that have selected is already reserved 2. They will be directed to select another seat
Special Requirements	- Java - Seat selection screen
Technology and Data Variations List	1a. A separate screen where users can see a map of the auditorium as well as what seats are available and which are already reserved before clicking whichever seat they would like to reserve 3a. A checkout page where the user can finalize their purchase after selecting a seat
Frequency of Occurrence	Almost continuously
Miscellaneous	- What colors should be used to represent seats that are reserved

	versus those that are still available? - When purchasing a ticket, should the user always be taken to the Seat Selection Screen, or only if they choose to specify a seat?

Use Case Section	*Comment*
Use Case Name	Logging out
Scope	Java-lagged team designed event ticketing software
Level	User Goal
Primary Actor	User
Stakeholders and Interests	- User: wants to log out of their account
Preconditions	- The ticketing software is running and executing normally - The user already has a valid account - The user is already logged into their account
Success Guarantee	- The user is successfully logged out
Main Success Scenario	1. The user navigates to the “settings” option 2. The user selects the “log out” option 3. The user confirms their selection and is logged out
Extensions	3a. The user confirms their selection

	but, is not logged out 1. The system will display an error to the user 2. The user will be prompted to attempt logging out again 3. The user will be given the option to submit a bug report to the Administrators
Special Requirements	- Java - Settings screen - Log out functionality
Technology and Data Variations List	1a. A page containing various miscellaneous settings for the user
Frequency of Occurrence	Almost continuously
Miscellaneous	- Will the “Log Out” button only be available within the settings page?

Use Case Section	*Comment*
Use Case Name	Deleting account
Scope	Java-lagged team designed event ticketing software
Level	User Goal
Primary Actor	User
Stakeholders and Interests	- User: wants to delete their account
Preconditions	- The ticketing software is running and executing normally - The user has already registered a valid account with the system

Success Guarantee	- The user's account is successfully deleted and removed from the system database
Main Success Scenario	1. The user navigates to the "settings" option 2. The user selects "Delete Account" 3. The user re-enters their login information to confirm their identity 4. The user confirms their decision 5. The account is deleted and removed from the database
Extensions	3a. The user attempts to login with incorrect login information 1. An error is reported to the user 2. The user is allowed to try and input their login information up to 5 times 3. If 5 attempts are exceeded, the user will be directed to reset their login information 5a. The account is not deleted and/or removed from the database 1. An error will be reported to the system administrators 2. An admin will need to enter the database and manually correct the error
Special Requirements	- Java - Account deletion functionality
Technology and Data Variations List	1a. A page containing various miscellaneous settings for the user

	5a. A database which stores account information that can be edited to remove deleted accounts
Frequency of Occurrence	infrequently
Miscellaneous	- Should the user be required to complete two-factor authentication as well before they can delete their account?

Use Case Section	*Comment*
Use Case Name	Submit a bug report
Scope	Java-lagged team designed event ticketing software
Level	User Goal
Primary Actor	User
Stakeholders and Interests	- User: wants to submit a bug report to the System Administrators - College of Charleston: wants to be made aware of any issues to fix them as soon as possible
Preconditions	- The ticketing software is running and executing normally - The user has some error that they wish to report
Success Guarantee	- The user successfully writes a bug report - The report is successfully logged and sent to the system administrators

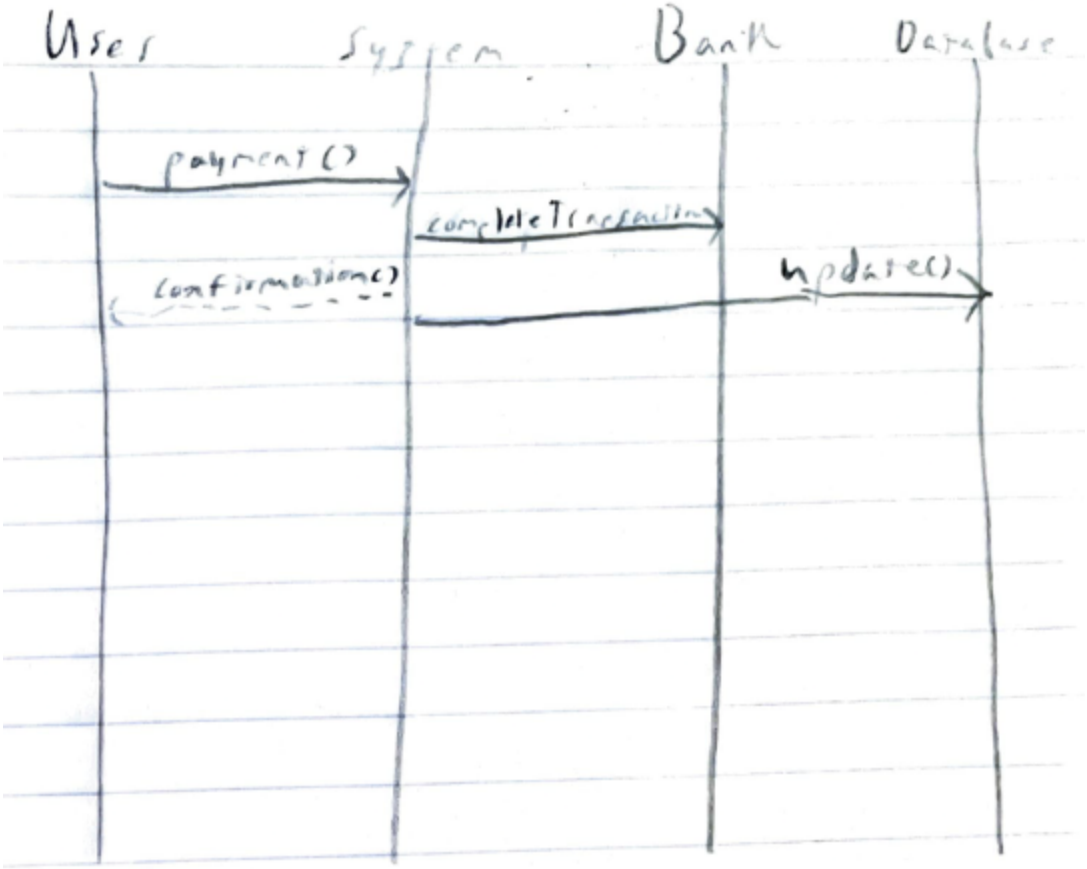
Main Success Scenario	1. The user navigates to the "Contact Us" section and opts to submit a bug report 2. The user is prompted to write a report with a maximum of 500 words, describing the error 3. The report is logged by the system and sent to the System Administrators
Extensions	2a. The user attempts to write a report longer than 500 words 1. The user will be notified that their report is too long 2. The user will be advised to try and explain their error in more concise detail 3a. The report is not logged or sent to the System Administrators 1. An error will be reported to the user 2. The report will attempt to be resent
Special Requirements	- Java - Bug reporting functionality
Technology and Data Variations List	1a. A page containing various miscellaneous settings for the user 2a. A Bug Reporting feature that would allow the user to write and submit a report of an error 3a. Database that can store these reports to be fixed by the System Administrators
Frequency of Occurrence	Rarely

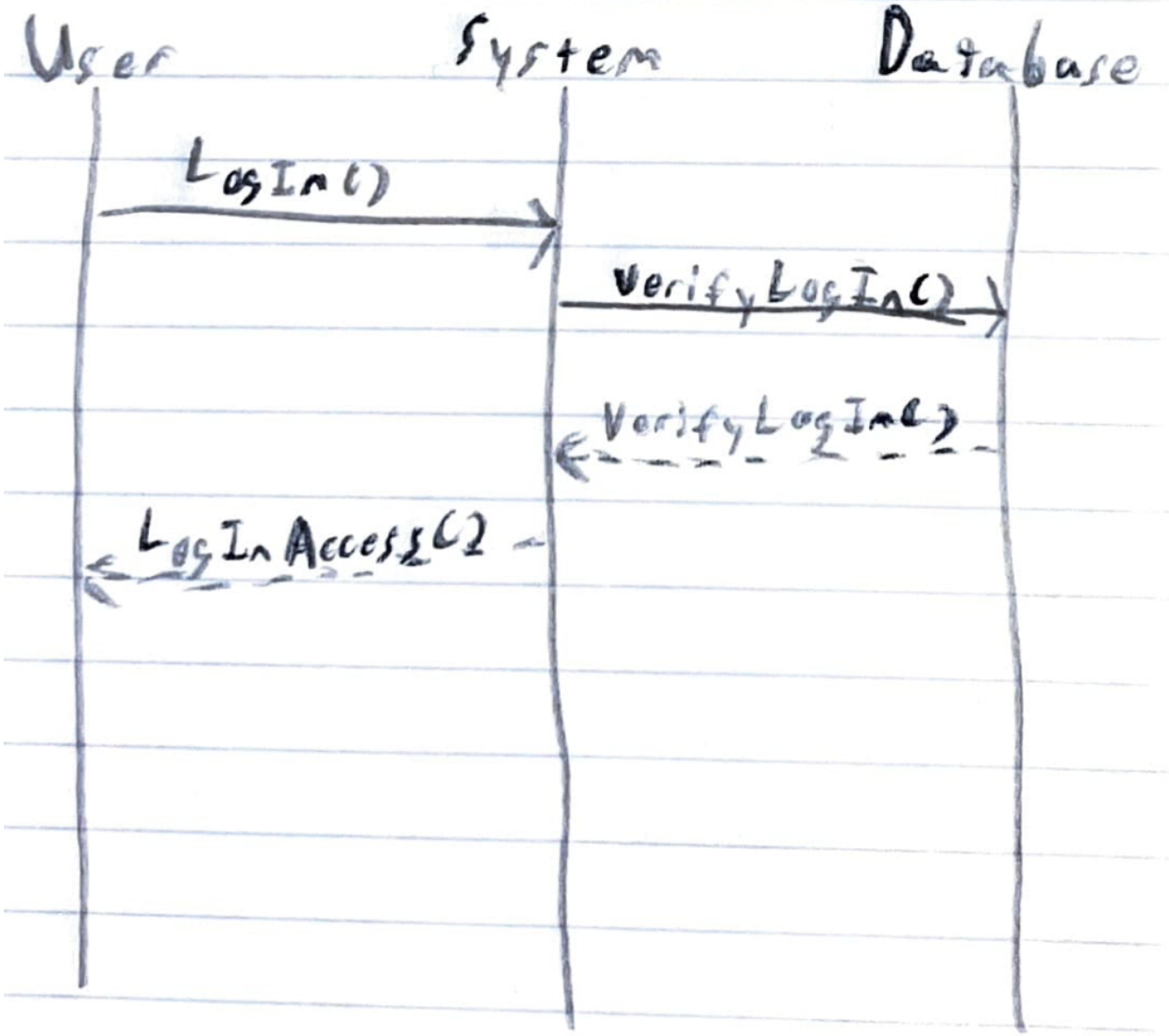
Miscellaneous	- Should the maximum word count be altered? - Would the “Contact Us” feature only be found under settings?

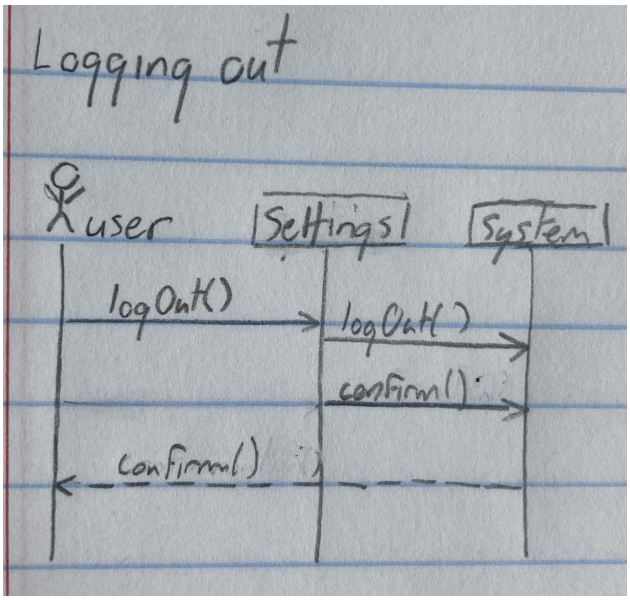
Use Case Section	*Comment*
Use Case Name	Selects a calendar event
Scope	Java-lagged team designed event ticketing software
Level	User Goal
Primary Actor	User
Stakeholders and Interests	- User: wants to use the interactive calendar to select an event
Preconditions	- The ticketing software is running and executing normally - There exists some events on the interactive calendar that can be selected
Success Guarantee	- The user selects an event on the interactive calendar - The user is given information on the selected event
Main Success Scenario	1. The user navigates to the event calendar 2. The user selects an event on the calendar 3. The user is taken to the description page for that event
Extensions	2a. The user is unable to select an event in the calendar

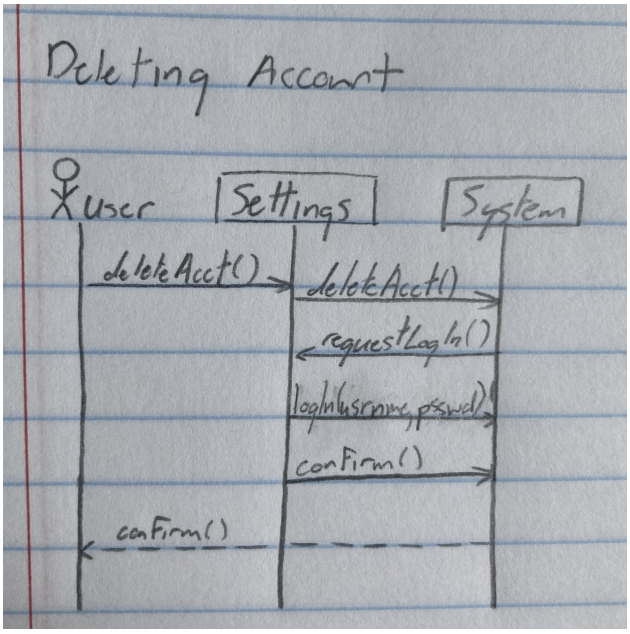
	1. The user will have the ability to submit a bug report about this issue 2. The System Administrators will be notified of the error and rectify it 3a. The user is not taken to the description page for the event 1. An error will be reported to the user 2. The user will be prompted to try re-selecting the event or selecting a different event
Special Requirements	- Java - Interactive Calendar
Technology and Data Variations List	2a. An interactive Calendar that displays events on a calendar depending on when they occur 3a. An event description page that provides information about the page as well as an option to select seats and purchase a ticket
Frequency of Occurrence	Often
Miscellaneous	- How many months in advance should the Calendar display events?

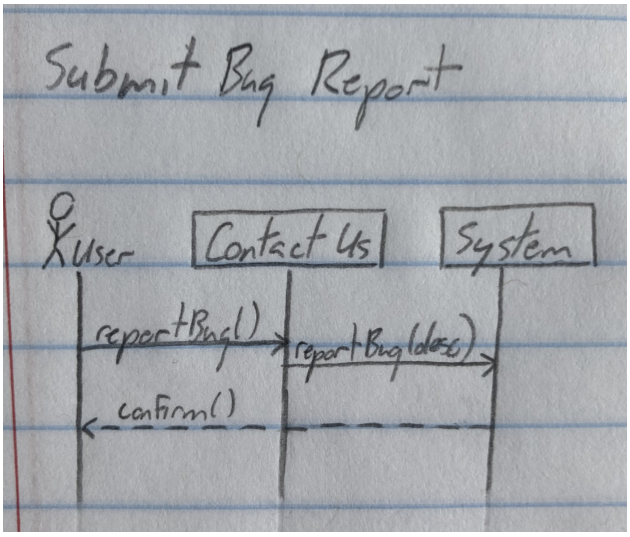
System Sequence Diagrams

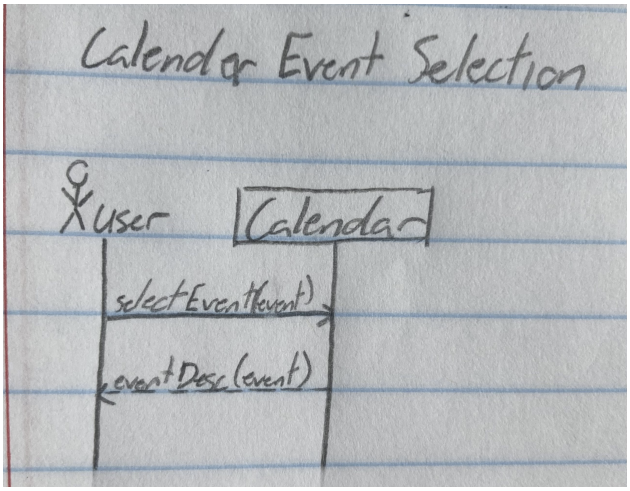

User
System
Bank
Database
payment()
completeTransaction()
confirmation()
update()


User
System
Database
LogIn()
VerifyLogIn()
VerifyLogIn()
LogInAccess()

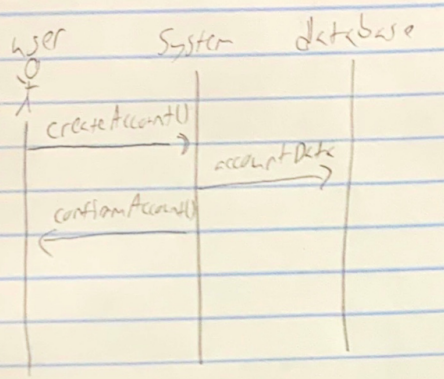

Logging out
user
Settings
System
logOut()
logOut()
confirm()
confirm()


Deleting Account
user
Settings
System
deleteAcct()
deleteAcct()
requestLogIn()
logIn(usrname, passwd)
confirm()
confirm()

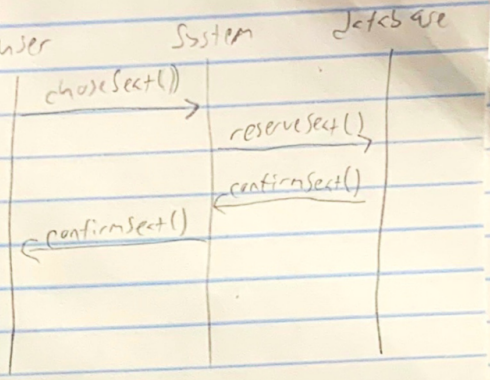

Submit Bug Report
User
Contact Us
System
reportBug()
reportBug(desc)
confirm()


Calendar Event Selection
User
Calendar
selectEvent(event)
eventDesc(event)

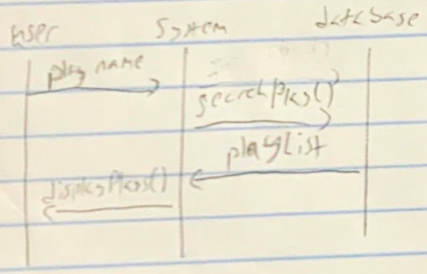
Create account



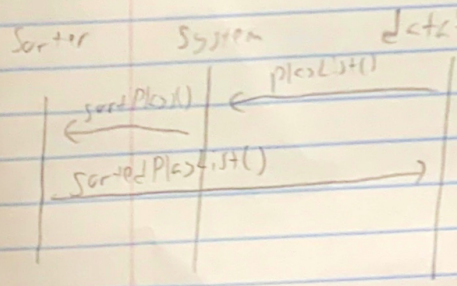
Selecting Seats



Secret for a play



Sorting Plays



Operation Contracts

Purchasing a Ticket	
Operation	payment()
Cross References	Use Case: Purchasing a Ticket
Prerequisites	The user has selected some event that they wish to purchase a ticket for
Post Requisites	Payment information is entered and ready to be processed

Purchasing a Ticket	
Operation	completeTransaction()
Cross References	Use Case: Purchasing a Ticket
Prerequisites	The user has confirmed their purchase
Post Requisites	The transaction is completed and recorded

Purchasing a Ticket	
Operation	update()
Cross References	Use Case: Purchasing a Ticket
Prerequisites	The transaction has been completed and recorded
Post Requisites	The ticket inventory has been updated with the new number of tickets

Purchasing a Ticket	
Operation	confirmation()
Cross References	Use Case: Purchasing a Ticket
Prerequisites	Payment is pending and ready to be processed
Post Requisites	Payment is either confirmed or denied by the user

Logging In	
Operation	logIn()
Cross References	Use Case: Logging In
Prerequisites	User has a valid account and is not already logged in
Post Requisites	The user has entered login information and is awaiting verification

Logging In	
Operation	verifyLogIn()
Cross References	Use Case: Logging In
Prerequisites	Login information has been input
Post Requisites	Login information is either confirmed or denied by the database

Logging In

Operation	logInAccess()
Cross References	Use Case: Logging In
Prerequisites	Login information has been checked by the database
Post Requisites	Login access is either granted or denied to the user

Logging Out	
Operation	logOut()
Cross References	Use Case: Logging Out
Prerequisites	User is logged in
Post Requisites	User confirms information

Logging Out	
Operation	confirm()
Cross References	Use Case: Logging Out
Prerequisites	- User is logged in - Logging out is underway
Post Requisites	User is logged out

Deleting Account

Operation	deleteAcct()
Cross References	Use Case: Deleting Account
Prerequisites	User is logged in
Post Requisites	User is prompted for login information

Deleting Account	
Operation	requestLogIn()
Cross References	Use Case: Deleting Account
Prerequisites	- User is logged in - Deleting account is underway
Post Requisites	Prompts User for login information

Deleting Account	
Operation	logIn(usrnme: String, psswd: String)
Cross References	Use Case: Deleting Account
Prerequisites	- User is logged in - Deleting account is underway
Post Requisites	User confirms information

Deleting Account	
Operation	confirm()
Cross References	Use Case: Deleting Account

Prerequisites	- User is logged in - Deleting account is underway
Post Requisites	- Account is removed from database - User is no longer logged in

Submit a Bug Report	
Operation	reportBug(desc: String)
Cross References	Use Case: Submit a Bug Report
Prerequisites	
Post Requisites	- Report is logged to the System - Report is sent to System Administrators

Submit a Bug Report	
Operation	confirm()
Cross References	Use Case: Submit a Bug Report
Prerequisites	A bug was reported
Post Requisites	A confirmation message is returned

Selects a Calendar Event	
Operation	selectEvent(event: Event)

Cross References	Use Case: Selects a Calendar Event
Prerequisites	Interactive event calendar is open
Post Requisites	Event selection is sent to system

Selects a Calendar Event	
Operation	eventDesc()
Cross References	Use Case: Selects a Calendar Event
Prerequisites	- Interactive event calendar is open - User selects event
Post Requisites	Description page for event is opened

Create Account	
Operation	createAccount()
Cross References	Use Case: Create Account
Prerequisites	An account can be created and stored in
Post Requisites	The transaction is completed and recorded

Selecting Seats	
Operation	chooseSeats()
Cross References	Use Case: Selecting Seats
Prerequisites	The user has selected a play that they wish to purchase a ticket for

Post Requisites	A ticket for an available seat or seats that the user has selected is generated and the user is taken to checkouts

Search for a Play	
Operation	searchPlay()
Cross References	Use Case: Search for a Play
Prerequisites	The user has navigated to the search bar and entered some criteria to search by
Post Requisites	Plays that match the criteria are displayed to the user

Sorting Plays	
Operation	sortPlay()
Cross References	Use Case: Sorting Plays
Prerequisites	There exists some plays in the Database that can be sorted
Post Requisites	The sorted plays are displayed to the user