



# SECURE PROGRAMMING AND SCRTIPTING

CA4

Killian Halpin (20093283)

Contents.....	0
Software System Requirements: .....	2
Log in:.....	2
Home Page:.....	2
Events Page:.....	2
Wallet:.....	3
Calendar:.....	3
NOTE: .....	3
Use Case Diagram: .....	4
Use Case Description: .....	4
Misuse Case Diagram:.....	5
Miss use Case Description: .....	6
Threat Tree:.....	6
DREAD Scores:.....	6
Analysis: .....	6

## Software System Requirements:

The software system I chose to do for this assignment is a Sports event registration system. The actors in this system would be the user. The user would be able to make and log into their accounts, sign up for sports events through the app, get notifications about new sports events added, pay for events which need payment and get news about events and past events in the system. The other actor would be the developer of the app, their interaction with the system would be making sure the system is up to date and in working order. Another actor would be the organiser of events. Their interaction would be to add events and pick their specific category for the events they added. They would also see a list of people who have signed up for the event and whether they have paid for the event or not.

### Log in:

#### **User/Organiser:**

For the log in page there would be an option for a user to make a normal user account or organiser account. When making the account the user would have to choose whether they are an organiser or a general user. When this choice is selected the sign-up page would be the same for both. It would ask for their name, address, email, phone and to create a password. It would also ask the actor what their preferred category of sport would be.

#### **Developer:**

The developer's job would be to make sure this page is working correctly, and the actors' details are being recorded properly so they can log back in if they ever log out. Faults with this page could include the actors log in detail not saving properly.

### Home Page:

#### **User:**

Once logged in the home page would be the first page seen by the user. On this page they could see all news about upcoming events, past events from all categories of sports. They could also see results from previous matches, races etc.

#### **Organiser:**

Once the organiser logs in they will have a different screen, there screen will look like a developer screen. From this screen they can add news about upcoming events with a description of each, add results from previous events and add pictures. They could also switch the user view which means they could see what the general user can see if they wish too. This would be where they add news about each event.

#### **Developer:**

It would be the developer's job to make sure this page runs correctly. They could also make it so that news about the favourite category comes up first for each specific user, so they see the news they are most interested in first.

### Events Page:

#### **User:**

The events page for the user would show all events in the specific category which they have selected. It would show dates times venues and a sign-up choice for the event. Some events would

need payment which could also be done through the system. The sign up for the event page would ask for their name, number, email address, address, and payment method if payment is needed. The user can pay through a wallet in the system, manual type in their details or pay through apple or google pay. There would also be a money back option if the user wanted to pull out of the event.

**Organiser:**

The organisers page event page would look slightly different. They would be asked to fill in detailed information about the event they are adding and send it to the developer so they can make sure it is a genuine event and not a made up one. Where the event is held, what the event is for (charity, fundraiser etc) date, time, and charge for the event. If there is a charge the money would go straight from the user's card/wallet into a wallet in the system set up by the organiser which he can easily withdraw funds from when needed. If user wanted a refund the organiser would get a notification which they would have to accept for the user to get their money back.

**Developer:**

It would be the developers' job to make sure this page is working perfectly. The forms would be sent to the organiser for all the users that sign up to events. The developer would also have to make sure the payment methods is secure so that user accounts cannot be stolen from. Also, must make sure the refunds function is working. It is also the developer's job to make sure events being added to the programme are real.

**Wallet:**

**User:**

The user would be able to add and withdraw money from the wallet. They can also pay for events with this wallet.

**Organisers:**

The organisers wallet would be filled with money from users signing up to events. They can withdraw money through the programme.

**Developers:**

It is their job to make sure this wallet is secure and safe for both users and developers, so their money is safe.

**Calendar:**

**User / Organiser:**

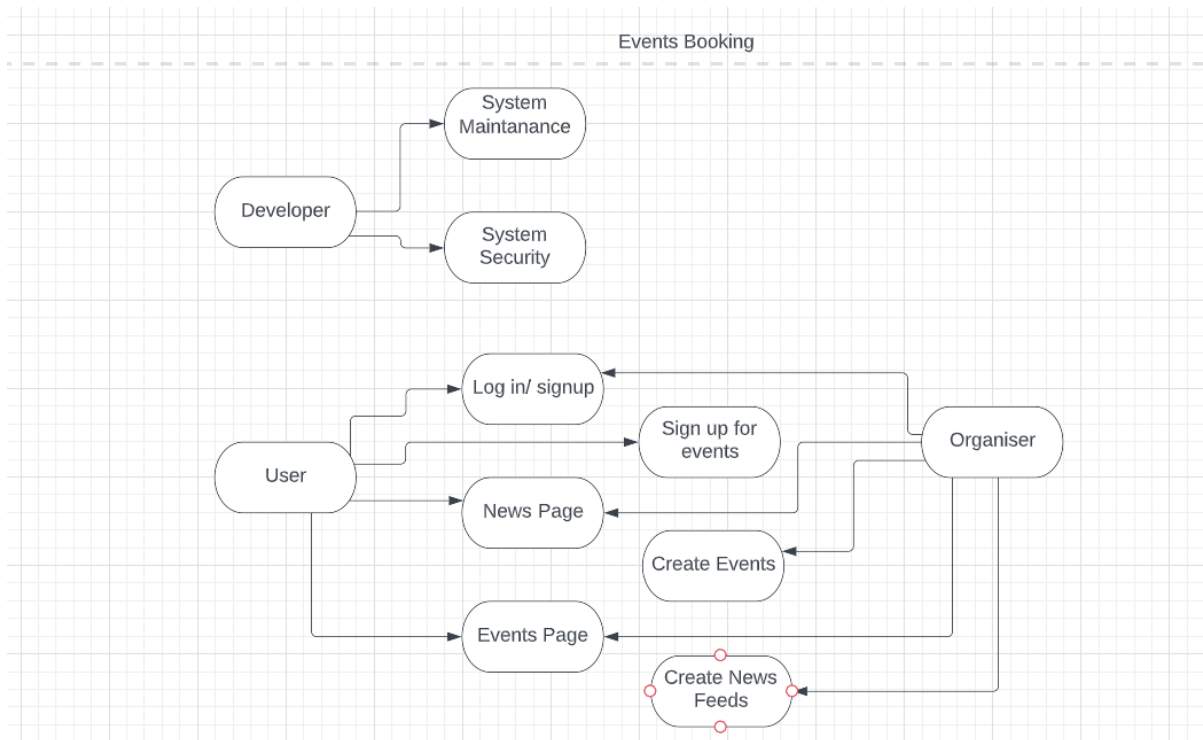
The calendar would be the same for both the user and the organiser it would simply just show what events they had coming up.

**Developer:** The developers' job would be to just make this feature is running correctly.

**NOTE:**

Personal PC requirements for this programme would be small. Mobile application could also be created.

## Use Case Diagram:



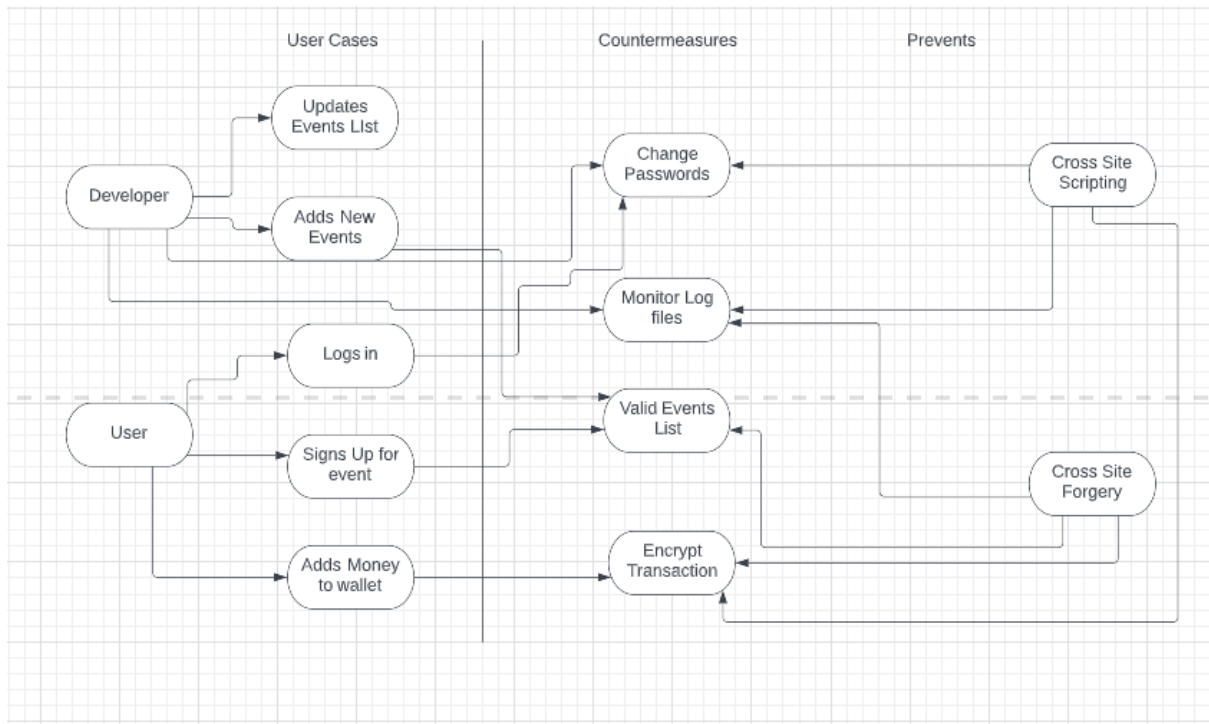
The developer is the only person who has access to updating the software and the security of the system. The user can log in or signup as can the organiser. However, the user is the only one who can book their spot in an event. The organiser is the only one who has access to create events and create news feeds.

## Use Case Description:

<b>Use Case:</b>	Sports Event Reservation
<b>Scenario:</b>	Online booking
<b>Triggering Event:</b>	User wants to book a spot in in an event and starts booking.
<b>Description:</b>	User searches for selected sports events. They check for availability in the event. They then book their spot in the event.
<b>Actors:</b>	User
<b>Related Use Cases:</b>	Creates individual account. Adds sports events they would like to see.
<b>Stakeholder:</b>	Student, Sports Event
<b>Preconditions:</b>	User account must exist Event must exist
<b>Postconditions:</b>	Booking must be created and linked with user Payment must be created with user Account.

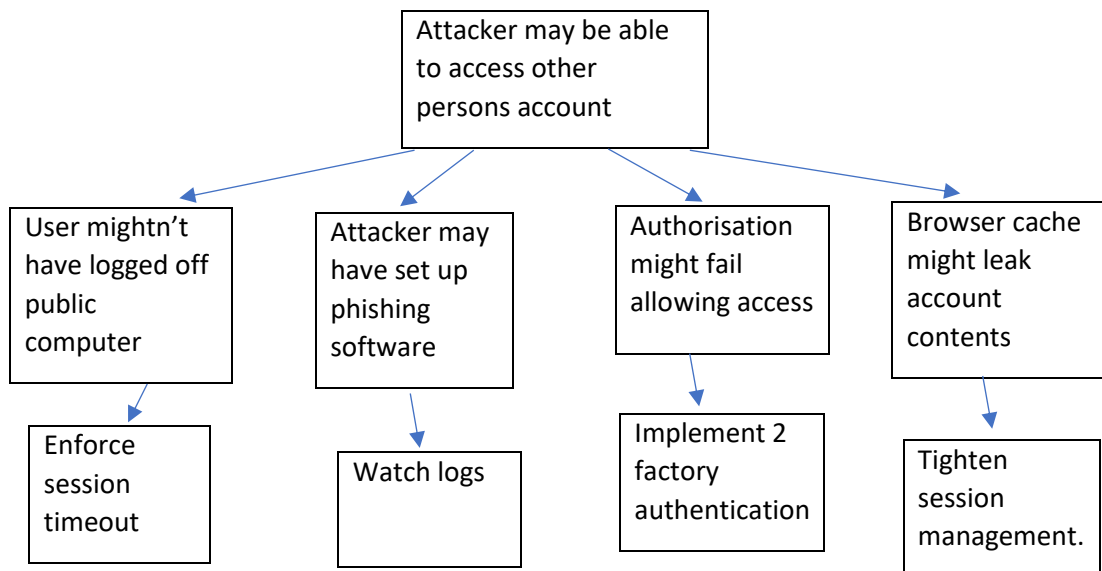
<b>Flow of activities:</b>	<b>Actor:</b> <ol style="list-style-type: none"> <li>1. Search for event</li> <li>2. Check availability</li> <li>3. Book event place</li> <li>4. Enter user details</li> <li>5. Enter payment method</li> </ol>	<b>System:</b> <ol style="list-style-type: none"> <li>1. Display Booking info</li> <li>2. Display availability</li> <li>3. Make reservation</li> <li>4. Send information to organiser</li> <li>5. Verify user info and payment info</li> <li>6. Create payment transaction</li> <li>7. Display Confirmation</li> <li>8. Send email confirmation</li> </ol>
<b>Exception Conditions:</b>	Payment method fails.	

### Misuse Case Diagram:



Miss use Case Description:

Threat Tree:



DREAD Scores:

Threat: Attacker views user personal information like name, address, and bank account details.

**Damage Potential:**

Reputation damaged and legal battle (9)

**Reproducibility:**

Totally reproducible (10)

**Exploitability:**

Attacker would have to be on the same subnet (6)

**Affected Users:**

Affected all users (10)

**Discoverability:**

Easily found out (10)

Analysis:

With this system there is a possibility that users could get attacked as personal and bank details are available on the system. If an attack was to happen it would be detrimental to the system as all users would immediately stop using the system and they would not be able to gain any new users. It is up

to the developer to keep this system secure. Storing personal information and the persons bank details in separate databases might reduce damage done if an attack was to happen. Hide user passwords as securely as possible and introduce 2 factor authentication to make it harder for attackers to gain access to peoples' account. With this system it would be quite easy to make a phishing site to gain peoples usernames and passwords. It is important the developer looks and the logs as often as possible. To evaluate the code to make sure it is secure you could run tests using focused static analysis tools. Get a peer to review the code and make sure it is ok, can be good to get a fresh pair of eyes to review the code. Could also send it to a penetration tester to make sure it is secure.