

The first step in establishing the requirements of a project is to identify the stakeholders involved. They are divided into two categories: the people who are impacted by the system and the ones who impact the system. In our case, the people who will be impacted by the system are the users. A user can be any person who has a B:Active membership, members of the clubs and societies, or someone interested in sport events. The higher-ups in the University of Bristol's B:Active program and the societies are the ones that will impact the system.



The next steps are to identify some “user stories” for each stakeholder and their flow steps. For our project we have identified the following ones:

1. For students and external users:

“As a user, I want to be able to view all available events in one place, so that it’s easier to find what I’m interested in.”

“As a user, I want to be able to view the details of the events, so that I can make an idea of what is going to happen in the event and decide whether I want to participate or not.”

“As a user, I want to be able to book an event, so that I can let the staff know I want to attend it.”

These user stories have the following basic flow:

1. Download the app
2. Browse through the events
3. Select an event that you may be interested in
4. Read the details of that event
5. Book the event

An alternative flow may be:

1. Download the app
2. Filter the events
3. Browse through them
4. Select an event
5. Read the description
6. The user aborts and the event is not booked.

2. For the higher-ups in the B:Active program:

“As the health and wellbeing department staff, we want to be able to register attendees for events, so that we can keep track of them easier.”

1. User attends event
2. They mark the event as attended on the app
3. The staff is able to see who has attended the event

“As the health and wellbeing department staff, we want to be able to manage events inside the application, so that we can add and delete events depending on user privileges (administrator/user).”

1. User books event
2. Event gets cancelled
3. Gym staff deletes the event from the app
4. The user receives a notification about the cancellation

“As the health and wellbeing department staff, we want to be able to receive feedback for our events, so that we know what to improve.”

1. Student attends an event
2. After the event is finished, they write a feedback for that event on the app

Exceptional flow:

1. Student books an event
2. They attend the event
3. They don't want to write a feedback, so they ignore it.

The story we have chosen to be the main one is: "As a user, I want to be able to view all available events in one place, so that it's easier to find what I'm interested in."

The next step is to specify the atomic requirements, which are then divided into two categories: functional and non-functional requirements.

For the main story, the atomic requirements are:

- "The user must be able to download the app."
- "The user must be able to browse through all the available events."
- "The user should be able to filter the available events."
- "The user shall be able to read the description of the available events."
- "The user must be able to book an event."
- "The user shall be able to cancel the booking for any event."
- "The user must be notified in case an event gets cancelled."
- "The staff should be able to receive feedback after each event."

Lastly, we need to identify the functional and non-functional requirements, written in natural language. The functional requirements are:

BOOKING-1.0 : The user shall be able to download an application in order to allow them to use all the facilities provided by the system.

BOOKING-2.0 : The user must be able to browse through all available events from the timetable, in order to allow them to decide which events they want to attend.

BOOKING-2.1 : The user should be able to filter the events by their interest, in order to be easier for them to find something that they would like.

BOOKING-2.2 : The user shall be able to read a description of any of the available events in order to help them decide if they want to attend the event or not.

BOOKING-3.0 : The user must be able to book any available event in order to let the staff know they want to attend it.

BOOKING-3.1 : The user shall be able to cancel a booking for an event in order to let the staff know they are no longer able to attend it.

CANCELLATION-1.0 : The staff must be able to cancel any event, in case the event won't take place anymore.

CANCELLATION-1.1 : The user must be notified in case the staff cancels an event that the user was going to attend, in order to let them know they are not able to host it anymore.

FEEDBACK-1.0 : The staff should be able to receive feedback from the attendees of their events, in order to allow them to improve their services.

The non-functional requirements are:

EFFICIENCY-1.0 : The application shall meet the agreed response time and capacity requirements.

USABILITY-1.0 : The application should be capable enough to support 100 000 users without affecting its performance.

USABILITY-1.1 : The application shall be able to accommodate increased users and volumes.

SECURITY-1.0 : The application must provide access only to legitimate users.