User Stories

Individual Project: Festival Connect

Name: Lucas Jacobs
Class: S-A-RB02
PCN: 490692
Student number: 4607368
Technical teachers: Felipe Ebert, Bartosz Paszkowski
Semester coach: Gerard Elbers

Table of Contents

Introduction

This document will show all the user stories for the FestivalConnect. User stories are a foundation for this agile project, it gives an elegant way of showing the functionalities of what the user wants. They can be useful to identify the needs and expectations of the stakeholders simply and understandably. First, we will discuss how to come up with a good quality user story. To do this we need to combine multiple things such as what criteria a user story needs to hold up against, how to properly set the definition of “done” of a user story, and how to also properly set these user stories throughout the sprint.

Quality of a User Story

First, to begin with, a user story will be used to briefly describe a simple feature told from the perspective of your users and customers. User stories are key components when it comes to an agile way of working, making concise, user-descriptive functionalities. Unlike normal requirements documents, where, you would just point out the functionality, user stories will prioritize collaboration and conversation over sophisticated documentation.

The most basic user stories are expressed with a simple sentence, see the following:

"As a [Persona], I [want to], [so that]"

To list down the structure,

- Persona: This refers to the person that wants the functionality.
- Wants to: This is where you will describe what kind of functionality the persona wants to perform.
- So that: Describing what the desire is that the user wants to seek from this functionality.

Therefore, we can rewrite the user story to:

"As a <User type>, I want <a goal> sot that <reason>"

(REHKOPF, n.d.)

Why Create User Stories?

User Stories can be written for both developers and clients benficial. They serve a way of having a communication aspect between developers and clients to understand what they want from each other. So a client can easy understand the user stories, therefore developers have less communication problems with clients. It is also good to have explain the function you want to implement to give it a certain context/action that it needs to achieve, so that it further clarifies the reason of having the function.

Writing a Good Quality User Story

When we work in an agile way, the scrum team often uses the INVEST principles to identify when a user story is of good quality so that it can be used as insight into the sprint. Every user story that is created, needs to be completed within a sprint timeline (three weeks). The user story must be manageable and testable, therefore see the following points of the INVEST principles and how this will make a good user story that fits within a sprint. (Thoughtworks, 2014)

- **Independent**: Each user story should be self-contained, and not overlap with other stories in other to be completed. So try to **not** create user stories that depend on each other, that can lead to a bottleneck and a delay in development.
- **Negotiable**: A user story needs to be open for discussion and negotiation between stakeholders and the development team, in my case discussing it with myself and the stakeholders (teachers). This way we can make certain adjustments based on feedback.
- **Estimate:** The user story needs to be clearly defined and has a clear scope, this way I can estimate the effort required for the implementation. Therefore, don't have a vague understanding of user stories and it is hard to give an estimated effort.
- **Small:** It should fit within the sprint, in my case three weeks. They should focus on delivering value to the users. Do not make a user story that needs to be spread over several sprints. This will lead to delays in delivering and tracking progress.
- **Testable:** The user story needs to have clearly defined acceptance criteria, to be tested and considered complete. This can be easily done by first doing the tests before developing the functionality.

(Thoughtworks, 2014)

Rightly Sizing Your User Story

Another thing to look at when making the user story is the right size. Both smaller and larger user stories have their advantages, this mainly depends on the team and preferences, but also the length of a sprint. (Kingdon, 2019)

First, let's list the advantages of having larger user stories:

- **No Challenges of Splitting**: Larger user stories may not be required to split into smaller tasks, saving down into not further breaking down stories.
- **Perceived Efficiencies**: It can be looked at as being more efficient because it is a bigger task.
- **Clear Business Value**: A Larger user story can give more background on a business value, making it easier for the stakeholder to understand.
- **Easier to Prioritize**: Larger user stories can be easily prioritized because they cover a bigger part of functionality.

When using bigger well thought user stories, can deliver value to users and stakeholders, while streamlining the development process.

For the smaller user stories, there are also advantages to point out.

- **Accurate Estimates**: Easier to estimate the time to spend, leading to more precise planning.
- **Planning Flexibility**: More flexible in planning, giving teams the allowance to adjust priorities and efficient based on changing requirements or circumstances.
- **Measure of Progress**: With a smaller user story, you can track the progress more effectively.
- **Understanding of Scope**: A small user story has its focus on something specific, so it is more clear for other stakeholders.

(Thoughtworks, 2014)

Now you can question, what will be the right way for me? There we can break down user stories as far as you want. This can rely on the agreed story size of the development, so for me setting up strict rules on how I want to size my stories. Also progressively think about breaking down stories, leaving room for change. Also, think of the complexity, the operation of the user story, and the data boundaries to guide you in making the right size of a story. To clarify, see the following picture.

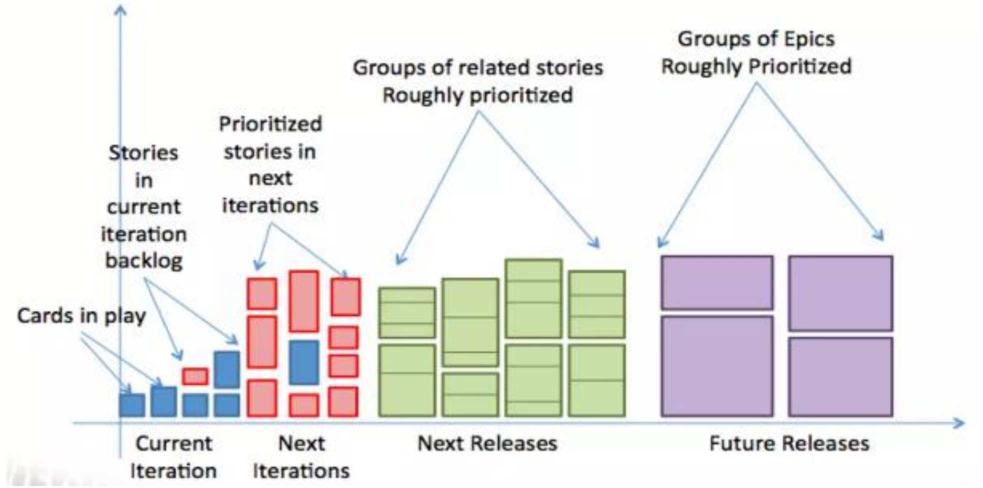

Figure 1: Breaking down user stories

Defining a User Story with Done

When writing a user story, we also need to know when we met these criteria of ‘done’. This is where the acceptance criteria come into play. It is a criterion that follows a certain format based on a given scenario. After this criterion is met, only the story can be seen as “complete”. The common format is as follows:

“Scenario: <Explain scenario>. Given <some initial context> when <an event occurs> then <ensure some outcomes>.”

To view what relation an acceptance test has in comparison to a User Story, see the following:

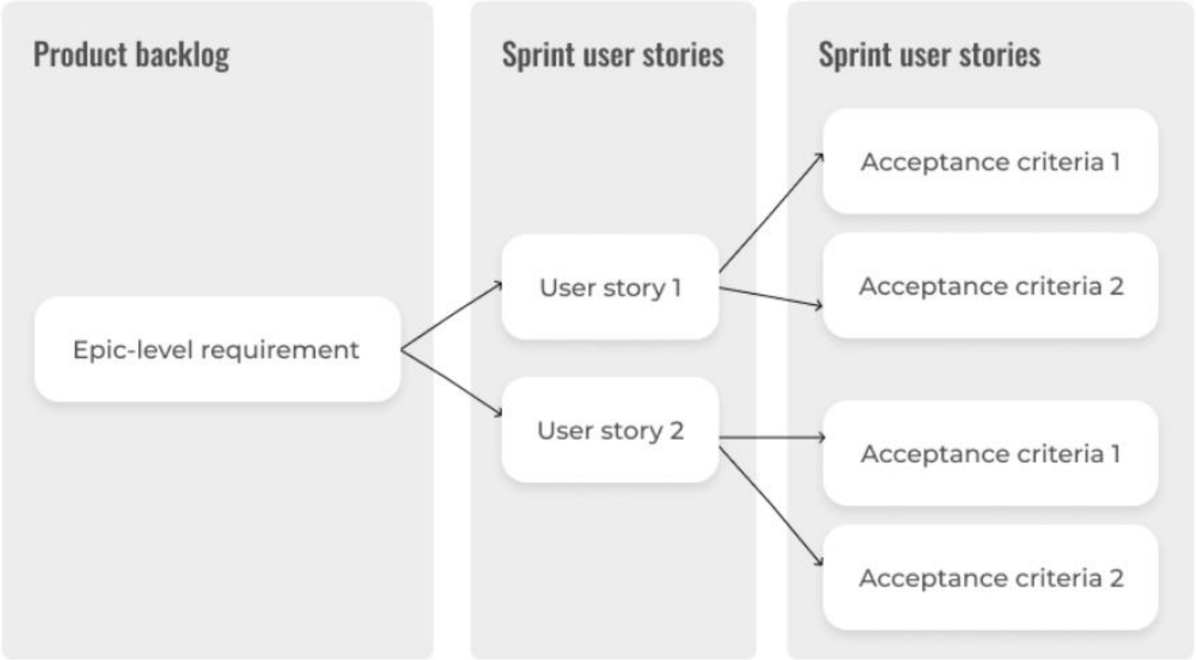

Figure 2 User Stories compared to Acceptance Criteria (ACCEPTANCE CRITERIA FOR USER STORIES, 2021)

To set a good standard of defining a User Story with acceptance criteria to the definition “Done”, we use SMART. To view why you need to be smart when doing acceptance criteria, let's go over them one by one.

- **Specific**: The criteria should clearly outline the exact condition that must be met for the user story to be considered “done”. This needs to ensure that everyone is aware of what needs to be achieved. What is the mission for your goal?
- **Measurable**: You need to have a look if the desired outcome has been achieved. Look at how you can determine this, making the goal more easy to have your head around.
- **Achievable**: It needs to be achievable within the context of the project, think of time, resources, and technical difficulties. So set realistic expectations of what can be accomplished given the available resources
- **Relevant**: It should be in sync with the overall project goals and objectives. Focus on bringing value to the users and stakeholders, by addressing their needs.
- **Time Bound**: Specify when the outcome needs to be achieved. Think of deadlines, milestones, or a certain time frame to deliver the expected results.

By applying SMART, we make sure the acceptance criteria will meet the expected goal of the user story, leading to achievable and well-defined project objectives.

(Write SMART Goals & INVEST for User Stories, n.d.)

User Story Mapping

We now know what the importance is of a user story, and how you make one with quality. The next thing is how are we going to prioritize the user stories. A simple way is to visually map your user stories out, you will break them down into parts, making sure nothing gets missed and this will result in a fluent experience. The process has several benefits:

- **Prioritize the user:** This approach focuses on the needs of the user, making sure that features are selected based on the value of the user. This makes sure that teams can prioritize the essential functionalities over less critical ones, making this a user-centric approach. In my case is set for myself the priorities by doing enough research on what the user finds the most important, but at the same time listening to my stakeholders.
- **Helps effective prioritization**: Seeing everything on a broader scale, will help you know what needs to be done.
- Breaks down complex tasks: Large tasks can be made into more small, and manageable user stories.
- **Delivers new value constantly**: In iterations you will prioritize the value of the user, to get better user feedback and learn from mistakes.
- **Builds team collaboration**: Promotes collaborative decision-making and team building with the team.

In my case, I am alone when doing this, so it will be much broader and stricter to myself, while trying to involve a bit of the stakeholders and the needs of the end user, by looking at what type of people they are. (MacKay, 2019)

Epics

An epic can be described as a group of user stories. All these stories are related to one another and combine to form a large story. An Epic can work across different teams and projects, and they can be labeled further together as a 'Theme". (A Complete Guide to Agile Epics, n.d.)

BackBone

With the backbone, you can show the customer journey from the beginning to the end, on a more high-level. This includes all the high-level activities the customer will complete while using FestivalConnect. Depending on how you use your backbone and story map, a potential way is to make it up of epics. This backbone is used to give the team the "why" of the journey. (Muldoon, 2023)

Properly setting up user story mapping

The user story mapping can be done in multiple steps. See the following 7 steps to create a good user story map:

1. **Frame the journey**: Make a common goal or vision for the product, how it looks, what problem needs to be solved, and the user benefits to the company.
2. **Build the story backbone**: Make the backbone, and give the whole user journey on high-level tasks/epics or steps from start to finish.
3. **Identify and group activities**: Look at common themes among the user steps and group them into activities, which can represent a goal.
4. **Breaking larger tasks into subtasks**: Divide the larger tasks/epics into smaller and easier-to-maintain subtasks.
5. **Fill in the blanks**: Test for missing tasks or another point of view by going threw the scenarios and outlining and understanding the user journey.
6. **Prioritize tasks and subtasks**: What are the most important tasks and subtasks, rank them on importance, considering the user value and dependencies.
7. **"Slice" groups of tasks into iterations**: Create horizontal "slices" that will make sure that you have a chronological release and outcomes, which will improve the iterative development. See the following picture.

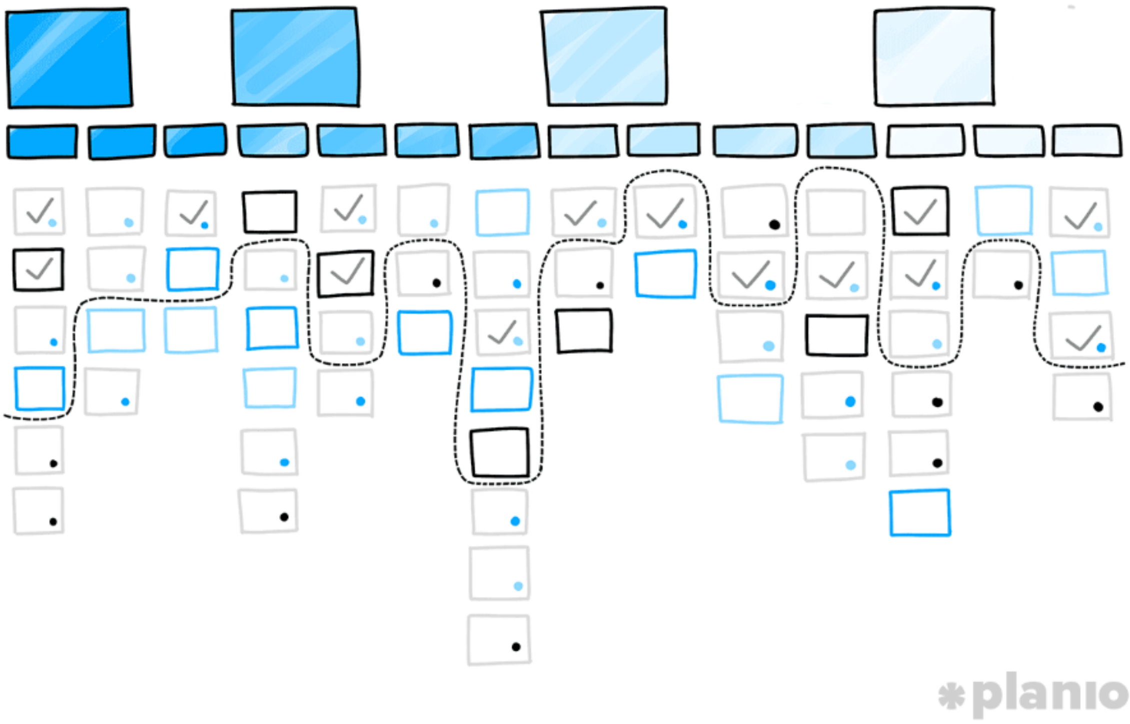

Figure 3: Ordering tasks

To conclude, user story mapping is a way to have a structured approach to making your product vision into a development strategy. With vertical slicing, prioritizing tasks during sprints will become easier and more effective. Therefore user story mapping serves as a dynamic tool for continuous planning and progress tracking, with the needs of what the company wants, in my case thinking of what the most important functionalities are for FestivalConnect.
(MacKay, 2019)

User Journey

Meet John, he is passionate about Hardcore music and attends Hardcore music festivals regularly. Joh wants to connect with other Hardcore music lovers and share his opinions and experiences through FestivalConnect.

Journey and Backbone

To further dive into it, John decides to start an account on FestivalConnect. His Journey at FestivalConnect begins with registering for an account. He will give his email and create a password to access the platform. To outline his whole journey, see the following backbone.

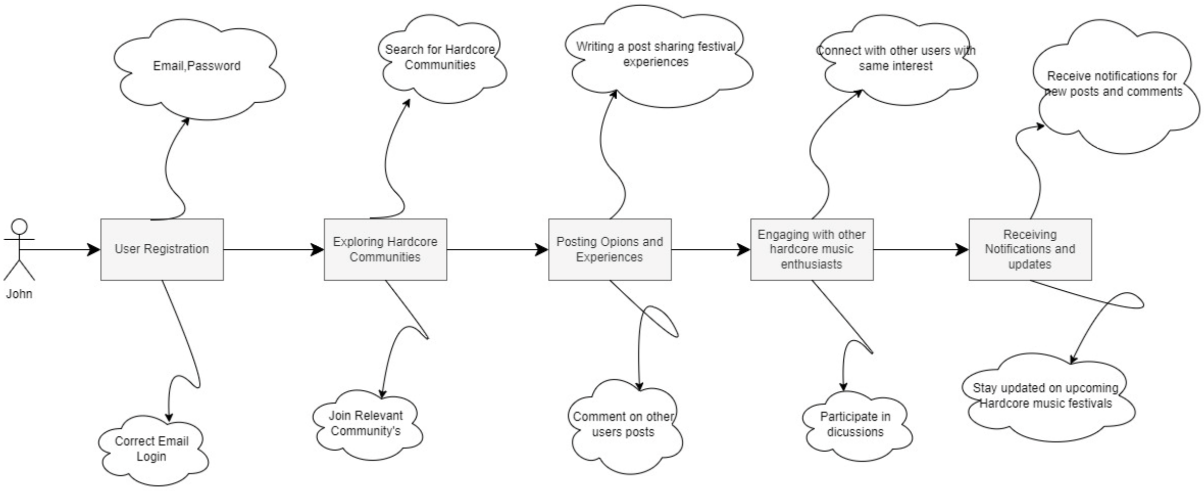
Email,Password
Search for Hardcore Communities
Writing a post sharing festival experiences
Connect with other users with same interest
Receive notifications for new posts and comments
John
User Registration
Exploring Hardcore Communities
Posting Opions and Experiences
Engaging with other hardcore music enthusiasts
Receiving Notifications and updates
Correct Email Login
Join Relevant Community's
Comment on other users posts
Participate in dicussions
Stay updated on upcoming Hardcore music festivals

Figure 4: Johns Journey with subtasks

Group Activities

We can separate this journey into groups, so we can focus on common themes and objectives to create a great overview.

Community Engagement and Interaction

- Activity 2: Exploring Hardcore Music Communities.
- Activity 3: Posting Opinions and Experiences.
- Activity 4: Engaging with Other Hardcore Music Enthusiasts.

These activities that John did are around the activities of his Hardcore genre festival communities on FestivalConnect

User Registration

- Activity 1: User Registration

In John his journey, this is the initial step to take to start your participation in FestivalConnect.

Notification Management and Stay Informed

- Activity 5: Receiving Notifications and Updates

This activity will try to make John stay informed about posts, comments, and updates within the Hardcore music festival communities he has joined.

Subtasks of Activities

For each of these activities, we can divide them into subtasks, such as:

User Registration	Exploring Hardcore Music Communities	Posting Opinions and Experiences	Engaging with Other Hardcore Music Enthusiasts	Receiving Notifications and Updates
Create user registration form UI	Design a user interface for exploring communities	Design UI for posting opinions and experiences	Design UI for viewing and interacting with community posts	Design notification settings interface
Back-end logic for user registration	Implement search functionality for finding Hardcore music communities	Implement a form for creating new posts	Implement functionality to display posts within a selected community	Implement logic to manage notification preferences
Validate user input during the registration process.	Display a list of available Hardcore music communities	Validate post content before submission	Enable the user to interact with posts (like, comment, repost)	Enable users to customize notification settings for different types of activities
Display confirmation message upon successful account validation.	Allow users to filter communities by genre, location, and popularity	Provide options for formatting the post and adding media	Implement real-time updates for new comments and interactions	Display notification pop-ups for new activities
Create login validation	Enable users to join communities of interest	Show previews of the post before uploading	Allow users to see other users there content	Provide the option to mark notifications as read or dismiss them
		Handle post submission and storage in the database.		

User stories

A user story needs to be created for every functional and non-functional requirement. A functional requirement will represent the functionalities that the user wants in FestivalConnect, and the non-functional requirements will say something about the behavior that the user expects from the product. All the user stories will have an ID and a certain format, with the importance mark based on the user story mapping.

Format

To have consistent user stories, we use the following pattern:

"As a [Type of user], I want [Function] so that [Action]"

The pattern doesn't need to be exactly followed as mentioned, but it is clear guidance to give a proper structure. Furthermore, the following roles will be used in the user stories.

- **End User**: This role represents the festival goers and the people who are joining communities and discovering festivals. They also have an option to make small communities for friends and like-minded people.
- **Festival organizer**: They can create their community, by sharing festival information.
- **Admin**: The people who manage the overall platform, such as user management and moderate user-generated content.

To properly number a user story, we want to use a structured format: "U-E001". The "U" stands for User Story, and "E" represents the type of user, in this case, the End user. The stakeholders are categorized as follows:

- E: End User
- F: Festival Organizer
- A: Admin
- N: Non-Functional Requirements

To further specify the workload, every week we work on FestivalConnect for three days, each day consisting of eight working hours. Therefore, each sprint will consist of 72 hours of workload (3 times (3 times 8)). To easily specify this, we will work with story points, each point is equal to four hours of workload. So each story point is equal to half a day of work, in total having sixteen story points to divide each sprint. This is guidance but can depend on the workload that needs to be finished for a sprint.

End User (E)

Story ID	User Story	Importance
U-E001	As an **End User**, I want to **login in**to FestivalConnect with my email and password, so that I can access personalized features and content	
U-E002	As an **End User**, I want to **register** as a festival-goer by providing basic information, so that I can create a profile and access FestivalConnect's features.	
U-E003	As an **End User**, I want to **view all available festival communities** on FestivalConnect, so that I can explore communities' interests.	
U-E004	As an **End User**, I want to **join festival communities** based on my interest, so that I can make conversations with like-minded people.	

U-E005	As an **End User**, I want to **post comments** in festival communities, so that I can engage in discussions and share my thoughts with other community members.	
U-E006	As an **End User**, who is interested in a specific genre, I want to filter the list of festival communities by genre, so I can easily find and select the festival that fits my taste.	
U-E007	As an **End User**, I want to **filter** the list of festivals **by date**, so I can easily find festivals that fit my schedule.	
U-E008	As an **End User**, I want to filter the list of **festivals by location**, so I can know if I can manage to go there.	
U-E009	As an **End User**, I want to **search by name**, so I can quickly locate and join specific festivals that I know.	
U-E010	As an **End User**, I want to **receive notifications** for new events, so that I can stay informed of the latest news of FestivalConnect.	
U-E011	As an **End User**, I want to have a **navigation bar**, so that I can quickly scroll through the application.	

Festival Organizer (F)

Story ID	User Story	Importance
U-F001	As a **Festival Organizer**, I want to **register** as an organizer on FestivalConnect, so that I can create and manage festival communities.	
U-F002	As a **Festival Organizer**, I want to **create a community**, so that I can get input on the festival.	

Admin (A)

Story ID	User Story	Importance
U-A001	As an **Admin**, I want to manage user information, so that I can make sure of the security of the accounts on FestivalConnect.	
U-A002	As an **Admin**, I want to permit **requests** of festival organizers, so that the user base can grow.	

Non-Functional Requirements (N)

Story ID	User Story	Importance
U-N001	As a **stakeholder**, I want to have **user stories** based on research and mapping, so that I have a clear understanding of what needs to be done.	
U-N002	As a **stakeholder**, I want to have a **project plan**, to have a clear outline of how the product is going to be done.	
U-N003	As a **stakeholder**, I want to **analyze various database frameworks**, so that developers can make an informed decision on choosing the best database framework for FestivalConnect.	
U-N004	As a **stakeholder**, I want to research **design principles** and best practices relevant to FestivalConnect, so that I can make informed design decisions to create a user-friendly application.	

U-N005	As a **stakeholder**, I want to **analyze the technical aspects** of FestivalConnect, so that I can provide how certain processes work.	
U-N006	As a **stakeholder**, I want to have an overview of the **version control system** for FestivalConnect, so that the development teams stay consistent and maintain a consistent working flow.	
U-N007	As a **stakeholder**, I want to have a **C4 diagram**, so that I know how FestivalConnect is structured.	
U-N008	As a **developer**, I want to have **prototypes**, so that the stakeholders have a clear overview of what they can expect from the design.	
U-N009	As a **developer**, I want to research **the front-end** framework, so that I can make the best decision to use the most suitable framework.	
U-N010	As a **developer**, I want to research **the back-end** framework, so that I can make a informed decision.	
U-N011	As a **developer**, I want to develop a **security plan** for FestivalConnect, so that I can ensure the security and integrity of the platform.	
U-N012	As a **developer**, I want to create a **DevOps document** for FestivalConnect, so that I can make sure the environment is efficient in development, testing, and deployment.	
U-N013	As a **developer**, I want to have an **acceptance test plan**, so that I can make sure that the product meets the stakeholder expectations.	
U-N014	As a **developer**, I want to generate a **test report**, so that I can provide the stakeholders with insight into the product's quality and readiness for release.	
U-N015	As a **developer**, I want to have a **test plan for user stories**, so that I can make sure the user story meets its functionality and is "done".	
U-N016	As a **developer**, I want to document **my progress of each sprint**, so that I can provide continuous improvement and give transparency to the stakeholders.	
U-N017	As a **stakeholder**, I want to have a summary of feedback from **Feedpulse**, so that I can understand the process and development of FestivalConnect.	
U-N018	As a **stakeholder**, I want to have a **task board**, so that I can properly plan and prioritize my tasks based on the scrum method.	
U-N019	As a **stakeholder**, I want at the end of the project a **reading guide** for FestivalConnect, so that I have a clear overview of the project.	
U-N020	As a **developer**, I want to give a rough estimate on the **performance of FestivalConnect**, so that there is a clear overview of what the load of the system will be.	
U-N021	As a **developer**, I want to have FestivalConnect handle up to **fifty million concurrent users** so that the system can handle spike traffic without performance issues.	
U-N022	As a **developer**, I want to have **clear guidelines** on how to implement standard requirements, so that the system is open for extension.	

U-N023	As a **developer**, I want to have a proper **errors handling mechanisms** so that the user can rely on the system for uninterrupted service.	
U-N024	As a **developer**, I want to set up a **microservice for user authentication**, so that users can securely access the system.	
U-N025	As a **developer**, I want to have a **setup of the front-end** in a environment, so that developers can start working.	
U-N026	As a **stakeholder**, I want to be **secured** from unauthorized users, so that users and sensitive data are safe.	
U-N027	As a **stakeholder**, I want to make sure that FestivalConnect supports **multiple languages** so that the usage of the app grows.	
U-N028	As a **developer**, I want to have a service for user management, so that ongoing maintenance and further progress can be made.	
U-N029	As a **developer**, I want to research which **architecture** fits FestivalConnect, so that I can make the most suitable decision.	
U-N030	As a **developer**, I want to set up a basic environment in **GIT**, so that I can have the environment to start working in.	
U-N031	As a **stakeholder**, I want to know what is going to be **implemented in FestivalConnect**, so that there is a clear overview of what is going to be done.	
U-N032	As a **stakeholder**, I want to know what **behaviors of the system** are crucial to keep in mind, so that FestivalConnect can keep track of their requirements.	
U-N033	As a **stakeholder**, I want to have a **research plan**, so that I have a structured framework to guide research efforts.	

References

A Complete Guide to Agile Epics. (n.d.). Retrieved from wrike: https://www.wrike.com/agile-guide/agile-epics-guide/

ACCEPTANCE CRITERIA FOR USER STORIES. (2021, 09 13). Retrieved from lanars: https://lanars.com/blog/acceptance-criteria-for-user-stories

Kingdon, D. (2019, 07 22). *The Magic of Small User Stories*. Retrieved from agilealliance: https://www.agilealliance.org/the-magic-of-small-user-stories/#:~:text=Throughput%3A%20Small%20stories%20allow%20testers,capacity%20is%20like%20a%20funnel.

MacKay, J. (2019, 01 14). *A Guide to User Story Mapping: Templates and Examples (How to Map User Stories)*. Retrieved from plan: https://plan.io/blog/user-story-mapping/

Muldoon, N. (2023, 06 20). *The Ultimate Guide to User Story Mapping [2023 Guide]*. Retrieved from easyagile: https://www.easyagile.com/blog/the-ultimate-guide-to-user-story-maps/

REHKOPF, M. (n.d.). *User stories with examples and a template*. Retrieved from atlassian: https://www.atlassian.com/agile/project-management/user-stories

Thoughtworks. (2014, 01 20). *How do you get more out of your User Stories?* Retrieved from slideshare: https://www.slideshare.net/ThoughtWorks/gettingmoreoutofuserstories-130405024009phpapp01

Write SMART Goals & INVEST for User Stories. (n.d.). Retrieved from paradigm: https://www.visual-paradigm.com/scrum/write-user-story-smart-goals/

Jacobs, L. (2024). Software Requirements Specification (Unpublished manuscript), FontysICT.