**NETCOMPANY – TRADING VISION PROJECT (TVP)**

**A0100 – ANALYSIS REPORT**

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

.

|  |  |
| --- | --- |
| **Version:** | 1.0 |
| **Status:** | Draft |
| **Approver:** | Ngô Thái Bình  Nguyễn Thị Diễm Trang |
| **Author:** | Ngô Gia Hân  Quách Hoàng Minh  Nguyễn Bảo Nguyên  Nguyễn Vũ Anh Thư |



**Document history**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Status | Comments |
| 1.0 | 30-11-2021 | Quách Hoàng Minh  Nguyễn Bảo Nguyên  Ngô Gia Hân  Nguyễn Vũ Anh Thư | Draft |  |
| 1.0 |  |  |  |  |

**References**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Reference | Title | Author | Version | Link |
|  |  |  |  |  |

**Contents**

**I**[**ntroduction**](#_3znysh7) **6**

[**Reading Guide**](#_2et92p0) **6**

[**Included Deliverables**](#_tyjcwt) **6**

[**Functional Epics**](#_3dy6vkm) **6**

[**Dashboard (Admin/Staff)**](#_1t3h5sf) **7**

[Overview of the event](#_ei5brdppedyf) 8

[Current event](#_990gltmet3sr) 9

[Upcoming events](#_1h666mls763) 12

[Create new event](#_8kevnqpkerka) 13

[Overview of the question bank](#_l6vsroh1hv71) 14

[Add a new question](#_ttp5kgc5ftce) 15

[View a question](#_bl4n5kc6j1ss) 16

[Approving a question](#_bl4n5kc6j1ss) 17

[Overview of the user information](#_l6vsroh1hv71) 17

[Overview of the rewarding](#_l6vsroh1hv71) 19

[Authentication / Authorization](#_u36am4wyagai) 22

[Authentication / Authorization](#_w7kb5z9mb9wn) 22

[Fail to sign in](#_nbzaaep3135m) 23

[Forgot your password](#_o7zjgaqjs3dj) 24

[Verification code incorrect](#_xnt0dycm0x2k) 25

[Reset Password](#_r3sviyi5wg8y) 26

[Confirm reset password unmatch](#_a1j2aw7lykfy) 27

[Dashboard (Player)](#_57sma3reqfsq) 28

[Main dashboard](#_avv3pug7nu40) 28

[Event details](#_r6zye6x8t5lt) 30

[User menu](#_j9sboak16whm) 32

[Contributing (Player)](#_x1xgj04zietb) 33

[Rewarding (Player)](#_ty0dx2bebym) 34

[Exchanging system (Player)](#_ktsmte7kecoe) 35

[Profile management (Player)](#_fu40lzg79wgm) 39

[Gameplay (Player)](#_agjtckoihbzm) 40

[**Standard Presentation Layer**](#_nmf14n) **44**

[**Graph Components**](#_1mrcu09) **44**

[Functional Description](#_cfof515s9o12) 44

[Line Chart](#_orpbctdb84nv) 44

[Bar Chart](#_tl6qtjcnxb4d) 45

[Stacked Bar Chart](#_trz9yfy0m3hi) 45

[Pie Chart](#_pz5jafqpzmqb) 46

[Embedded Google Map](#_111kx3o) 46

[Integration Design](#_y6zb93i586bi) 47

[Data Validation Rules](#_98t2rthzpyje) 47

[**Banner/Alert Component**](#_1egqt2p) **47**

[Functional Description](#_deqyu4xolsgc) 47

[**Dropdown Selection Component**](#_3ygebqi) **48**

[Functional Description](#_hdfcls5y3f8z) 48

[Table Component](#_3xk2aubwif7i) 48

[**Non-Functional Requirements**](#_1302m92) **48**

[**Performance**](#_3mzq4wv) **48**

[Capacity](#_tl82i7ft72mg) 48

[Availability](#_hvlgmhkbcnde) 48

[Usability](#_1rzv84ggdrui) 49

[Security](#_tn815unwkkj5) 49

[**Visual Expression**](#_2250f4o) **50**

[Mobile Friendly](#_9ttzj9k0x3jz) 50

[Responsiveness](#_6dct50n4jtqo) 50

[Performance on Mobile Devices](#_djqb2trgbpzr) 50

[Supported Browsers](#_haapch) 50

[Design Guideline](#_lnomyv17nuv7) 50

[**High-level Software Architecture**](#_319y80a) **50**

[**Architectural Goals and Constraints**](#_1gf8i83) **50**

[Architectural Model](#_40ew0vw) 50

[Model](#_ecld723cm934) 51

[View](#_8z78f9qxl4os) 51

[Controller](#_7nsc8fbhrsnl) 51

[**Development Principles**](#_1tuee74) **52**

[**Deployment Perspective**](#_3s49zyc) **53**

[Build Process](#_z0agbbjdj57y) 53

[Release Process](#_w78tsw9qv33p) 55

[**Performance & Technical Documentation**](#_45jfvxd) **56**

[Technical Documentation](#_guc1pk1ovl9g) 56

[D0180 – External Interface Design](#_8b9i5tbluofp) 56

[O0400 – Technical Infrastructure](#_8xjyvr7fkp62) 56

[O0500 – Software Architecture](#_hlva7obcgity) 56

[Performance](#_y8yo4mhi0ec7) 56

[**Security & Compliance**](#_2koq656) **56**

[Security Measures](#_ypl2bxgmkubf) 57

[Compliance to Legal Obligations](#_6lw2p2i18pzs) 57

[Authentication](#_mhwtbnr3uxta) 58

[**Operation and Hosting**](#_3jtnz0s) **58**

[Encryption of Data](#_5bdo4qjopw7w) 58

[Backup and Restore](#_ssuq6zrjxpzk) 59

[**Integration Strategy**](#_1yyy98l) **59**

[**Glossary of Terms**](#_4iylrwe) **59**

[**Technical Infrastructure**](#_rjefff) **59**

[**Environments**](#_1qoc8b1) **59**

[Development](#_330b12asz3si) 60

[Quality Assurance / Test & Production](#_8hlzixg3a1qd) 61

[**Release Plan**](#_14ykbeg) **62**

[**Project Roadmap**](#_3oy7u29) **62**

[**Changes**](#_338fx5o) **63**

[Changes in Requirements](#_n78sloqejd9h) 63

[Change Requests](#_irg0khwshris) 63

[**Project Dependencies**](#_1idq7dh) **64**

[**Risks**](#_42ddq1a) **64**

[**Outstanding Issues**](#_2hio093) **64**

[**Appendix**](#_wnyagw) **68**

[**Project Plan**](#_3gnlt4p) **68**

[**Requirements**](#_1vsw3ci) **69**

# Introduction

As a conclusion of the clarification phase of the project, this report serves as a consolidation of Trading Vision Project (TVP) original requirements, the development team’s proposal and the dialogue about every single requirement and acceptance criteria throughout the project timeline.

Based on the clarification report, a high-level design and corresponding acceptance criteria are produced during the design phases. (chỉ khúc design luôn hay chỉ cả dự án)

## Reading Guide

(hướng dẫn đọc report)

There are ……… sections in this report:

* The clarification report starts by establishing the different areas of functionality in the Functional Epics section. These epics are based on initial project description and objectives from the product owner and end users perspective and are composed of either specialized components or components described in the **Standard Presentation Layer** section.
* Requirements of a non-functional character, e.g. performance, usability and visual expression are described in the Non-Functional Requirements section.
* The architectural model, development principles, deployment perspective, technical documentation, security and compliance, operation and hosting topics are described as different perspectives under the section High-level Software Architecture. The Integration Strategy section gives an overview of integrations, while the Technical Infrastructure section clarifies the dependencies to .com and the logical and execution environments.
* To conclude, the Release Plan section provides an updated project roadmap (can be updated throughout the project), maps identified project dependencies and list of initial risks and issues.

## Included Deliverables

(trình bày deliverables của toàn bộ dự án hay của phases nào) có cần những doc nào nữa ko

In the clarification phase multiple project deliverables are to be delivered. In the clarification report the deliverables listed below are embedded either as isolated sections or in their relevant context, e.g. the User Interface Prototype deliverable is split across relevant functional descriptions, while the High-level Software Architecture deliverable has its own dedicated section.

* A0140 - Functional Epics
* A0150 - User Interface Prototype
* A0200 - Business Processes
* A0130 - Non-functional requirements
* D0170 - Conversion Strategy
* D0180 - Integration Strategy
* O0400 - Technical Infrastructure
* O0500 - High-level Software Architecture
* P0100 - Project Roadmap
* Control Flow Diagram

# Functional Epics

(mô tả requirement ntn? Cần viết nhũng ý nào? )

The following sections contain the identified functional epics. The epics are based on the perspective of the player and the staff that use the NCP system.

## Dashboard (Admin/Staff)

The Dashboard's main purpose is to provide an overview of events, users, rewarding, question bank information and some statistical data for the admin/staff.

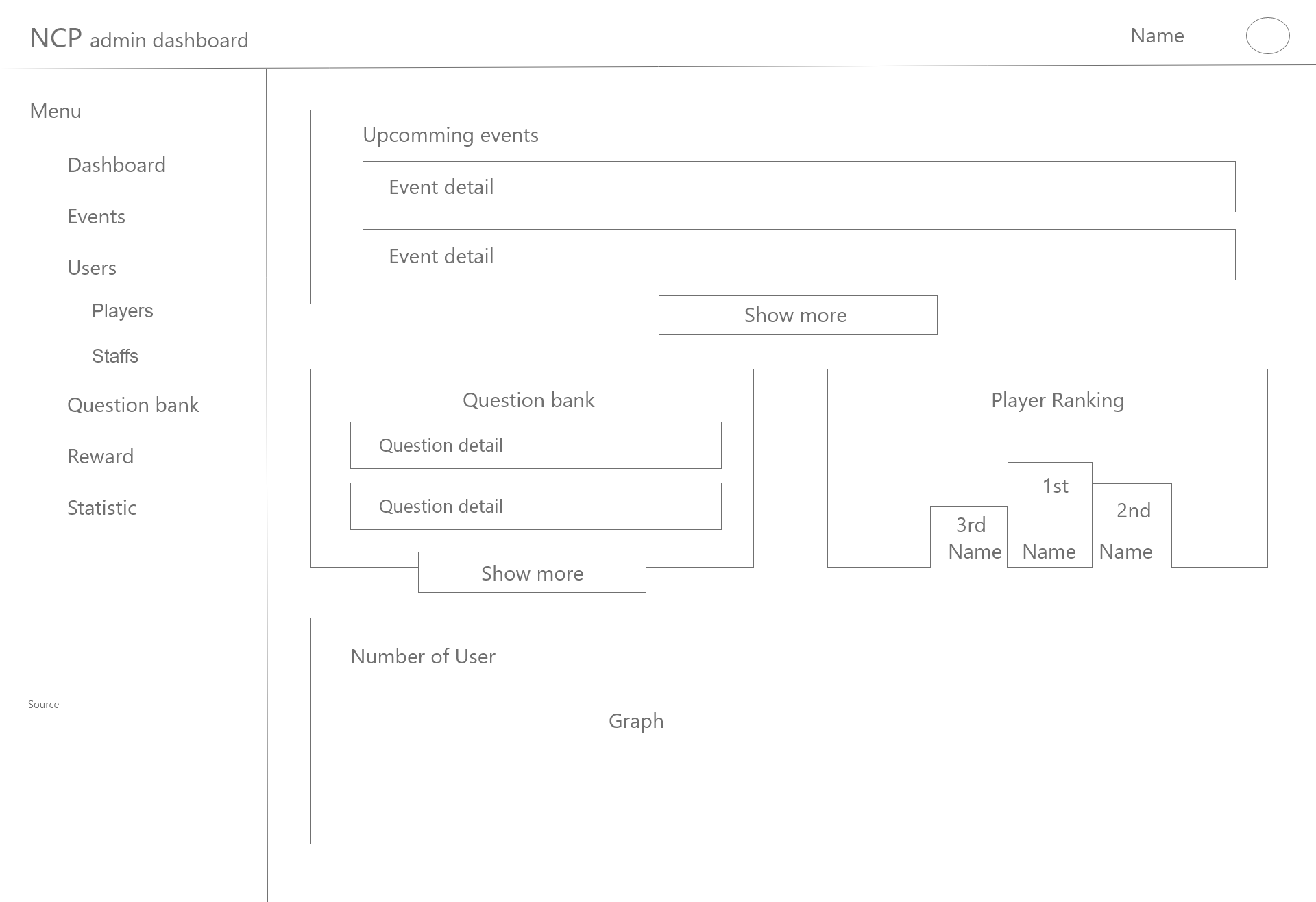


Figure 1 - Dashboard wireframe

As shown in Figure 1 the Dashboard will consist of multiple components. The top left corner will be the Logo of NCP admin Dashboard. On the left of the screen, there will be a navigation bar which leads the current user information to the corresponding feature section.

In the main container of the screen, the summarize information of upcoming events, questions bank. Click on each one will lead the current user information to the corresponding feature section.

The top right corner will be the name and the avatar of the current user. Clicking on the avatar will pop up the current user personal section.

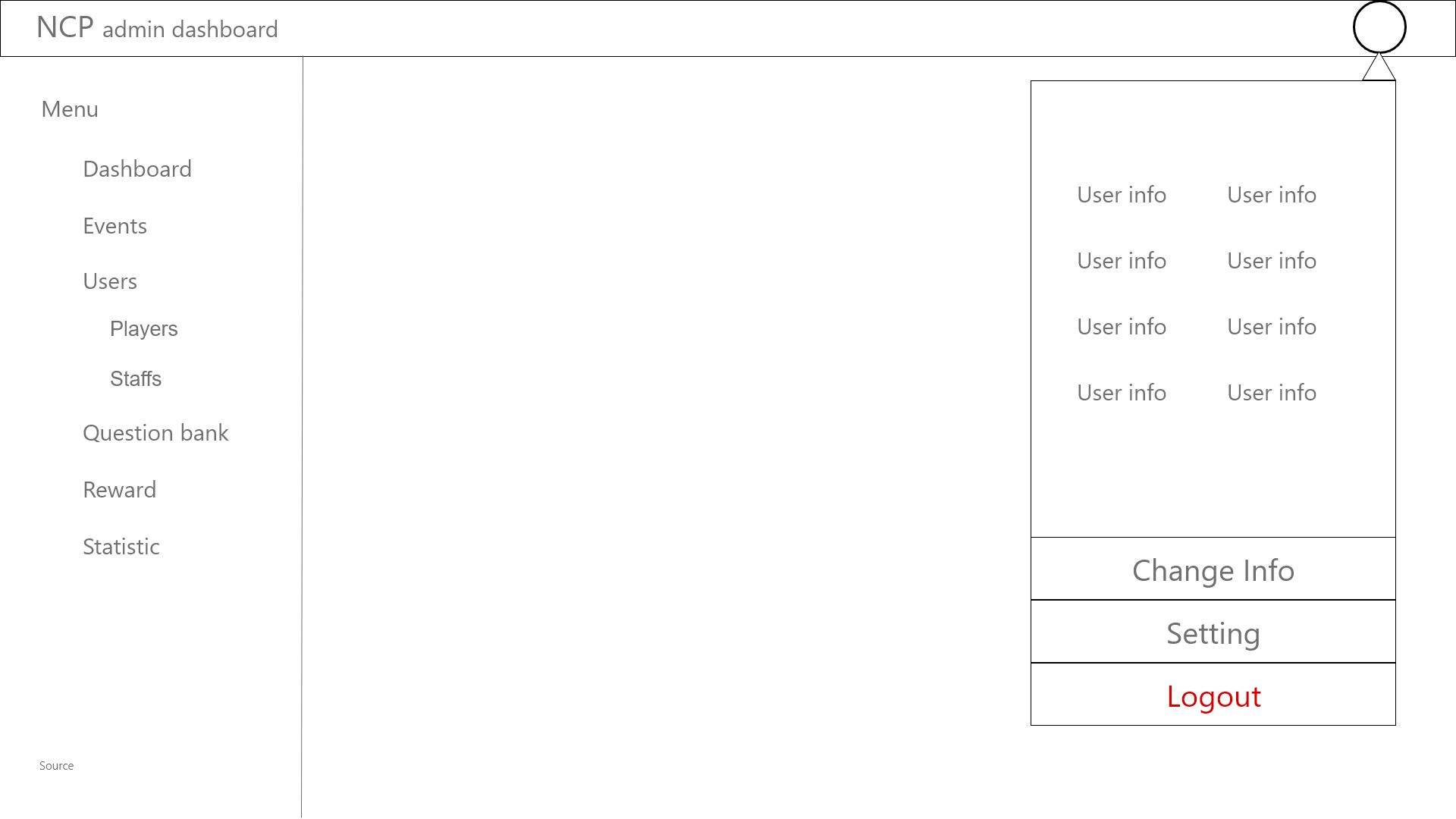


Figure 2 - Dashboard - Personal Section Wireframe

The personal section contains the current user information, information management feature and a logout option. (Figure 2)

### Overview of the event

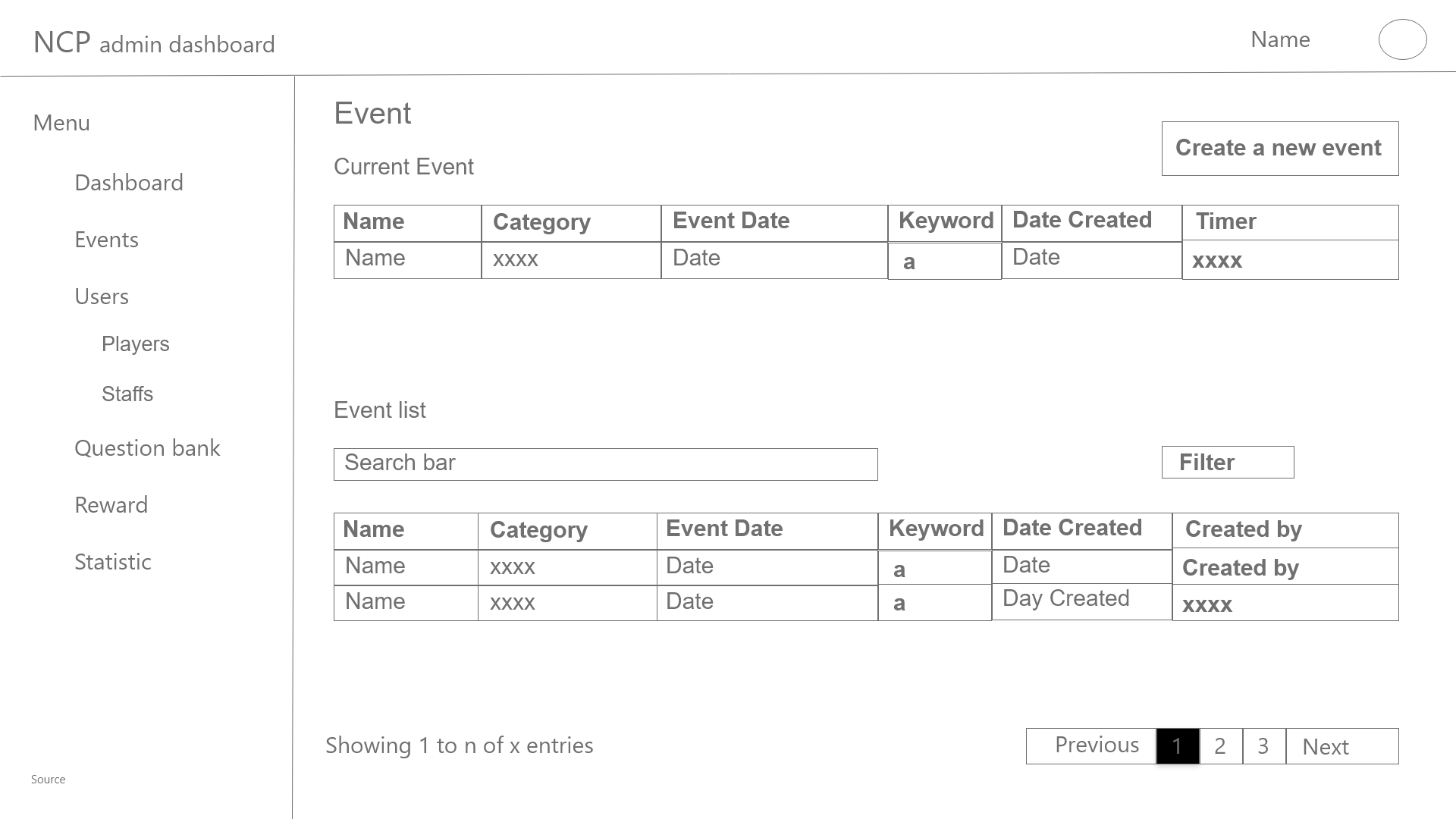
****

Figure 3 - Dashboard - Event wireframe

The layout (Figure 3) displays the general list of all of the events. As the figure shows, we can see that the page consists of two main parts.

The first part is about the ongoing event which is listed in detail (ID, name, category, date and time start, date created, core keyword and the remaining time of the event). It also contains a button that leads to the create a new event section.

The second part is the list of all events of the system. The events are listed with their information (ID, name, theme, date and time start, date created, core keyword and created by who), if the list is long enough, it will be automatically splitted into many pages.

The detailed page of the event that contains editable fields.

#### Current event

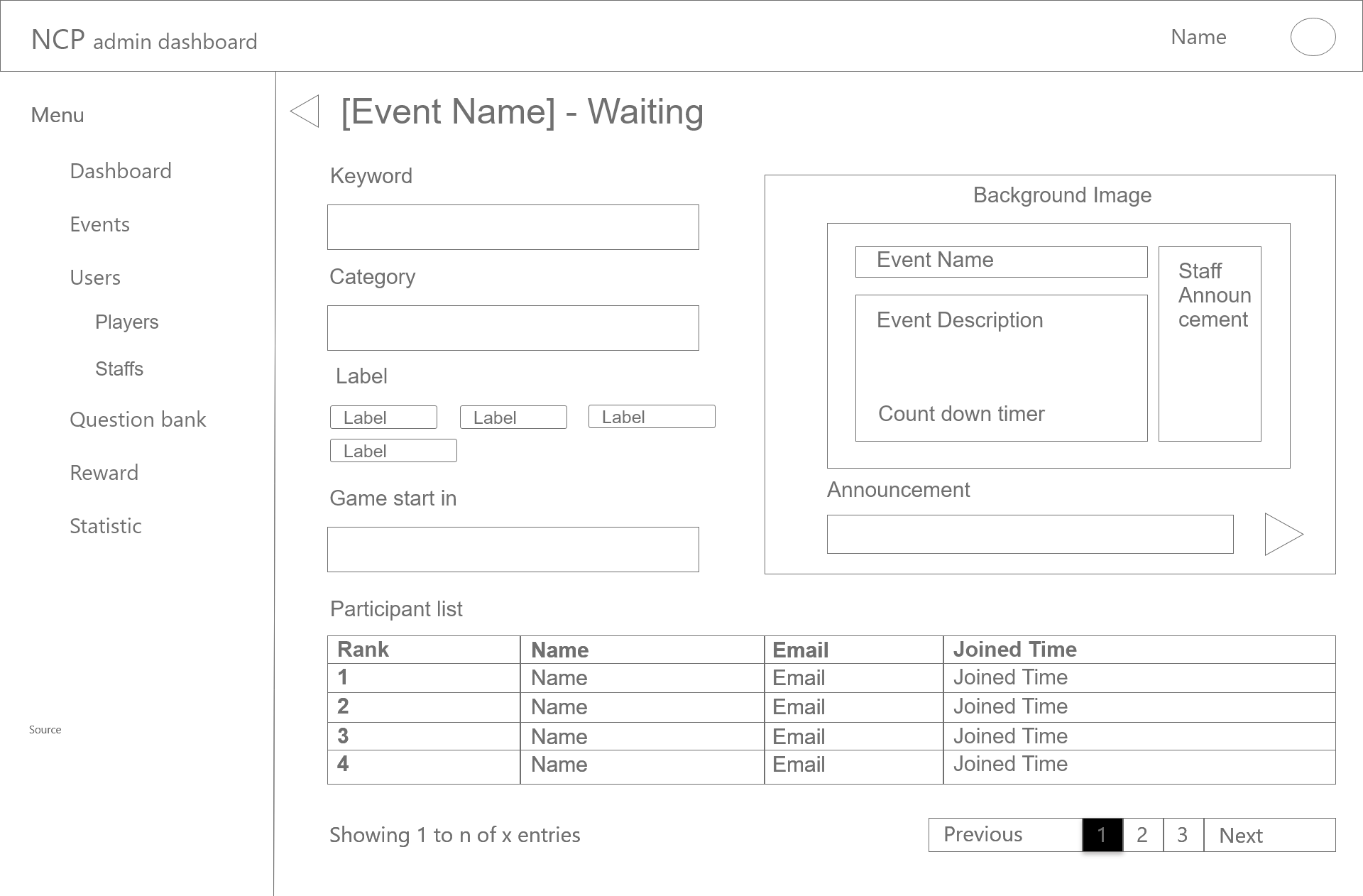
****

Figure 4- Dashboard - Event - Current event waiting room wireframe

After clicking on the current event in the event list, the waiting room page of the event will appear (Figure 4). This layout is the detailed view of the current event.

The main container of the page will consist of the data information(event name, keyword, category, label, start time, and participant list, the participant list will contain the rank, email, name and joined time, if there are many participants, the list will be splitted into pages). The top left corner of the container will have a back button that would lead the user back to the previous layout.

The right part of the main container there will be a small container which is a preview as a player perspective. The small container includes the event name, event description, countdown timer, staff announcements and staff announcement input field and send button.

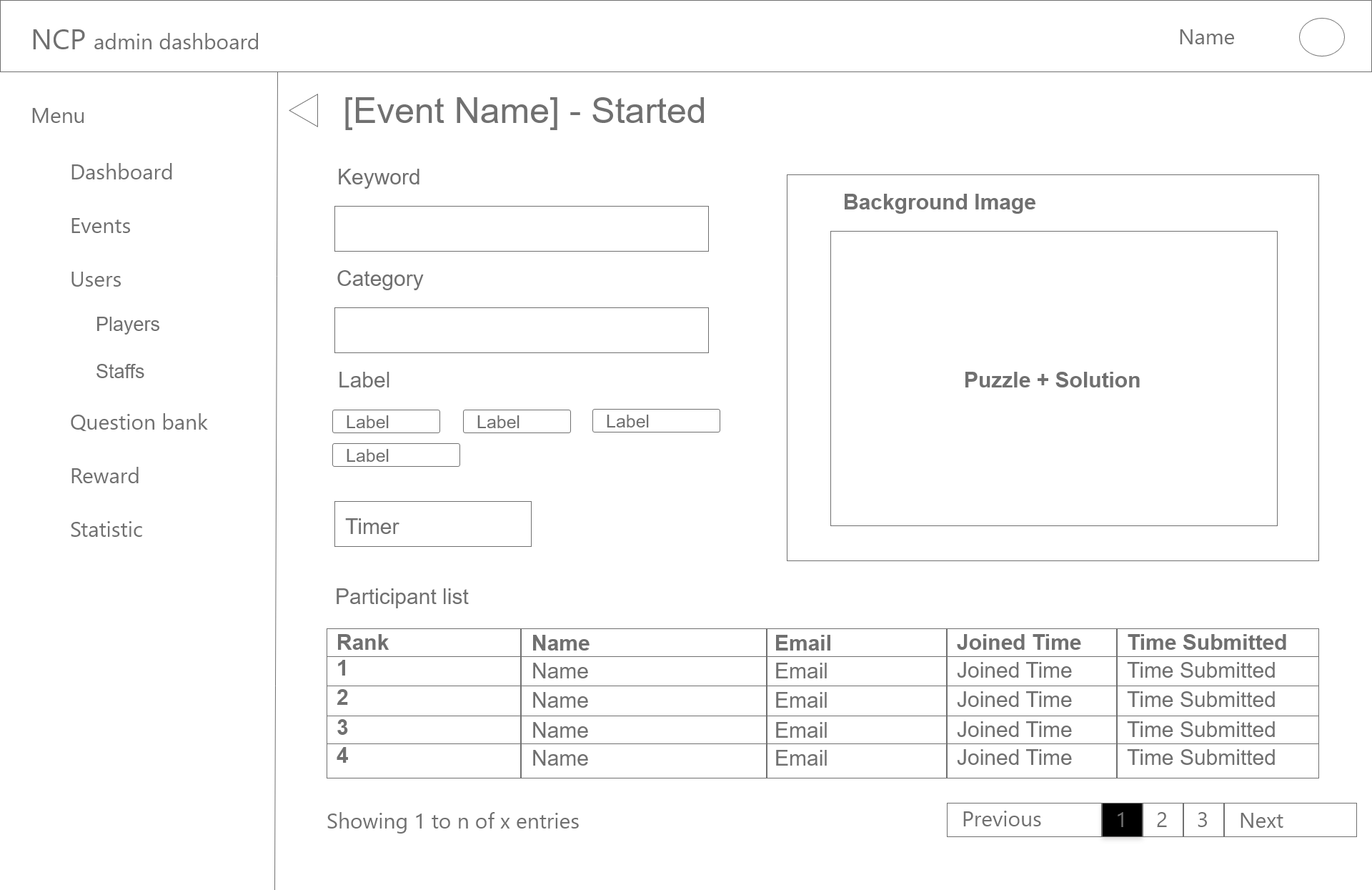
****

Figure 5 - Dashboard - Event - Current event started room wireframe

When the event starts, the waiting layout becomes the starting layout (Figure 5). This layout (Figure 5) will have almost the same information as the previous layout (Figure 4), but there will be a timer to show the time left of the event instead of the game start in timer. The participant list will have 1 more column, that is the time submitted which indicates the time that the players submitted their answer and the joined order will become the player’s rank in the event. The top left corner of the container will have a back button that would lead the user back to the previous layout.

After the time is up, the page layout (Figure 5) will change to the finished event layout.

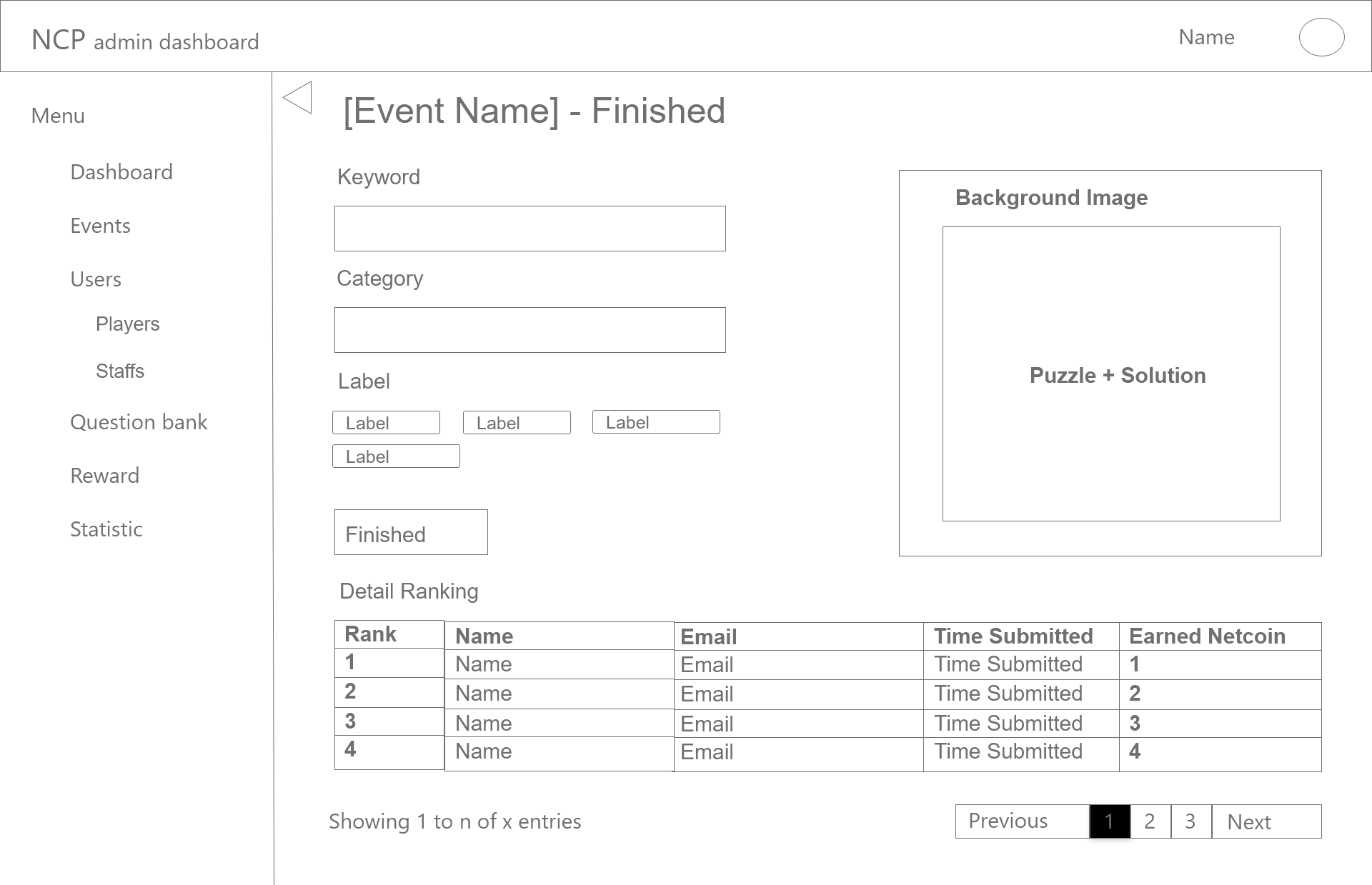


Figure 6 - Dashboard - Event - Current event finished room wireframe

This layout (Figure 6) will have almost the same information as the previous layout (Figure 5), but in the participant list, instead of displaying the joined time it will display the rank of the player and also it has one more column that is the earned Netcoin. The top left corner of the container will have a back button that would lead the user back to the previous layout.

#### Upcoming events

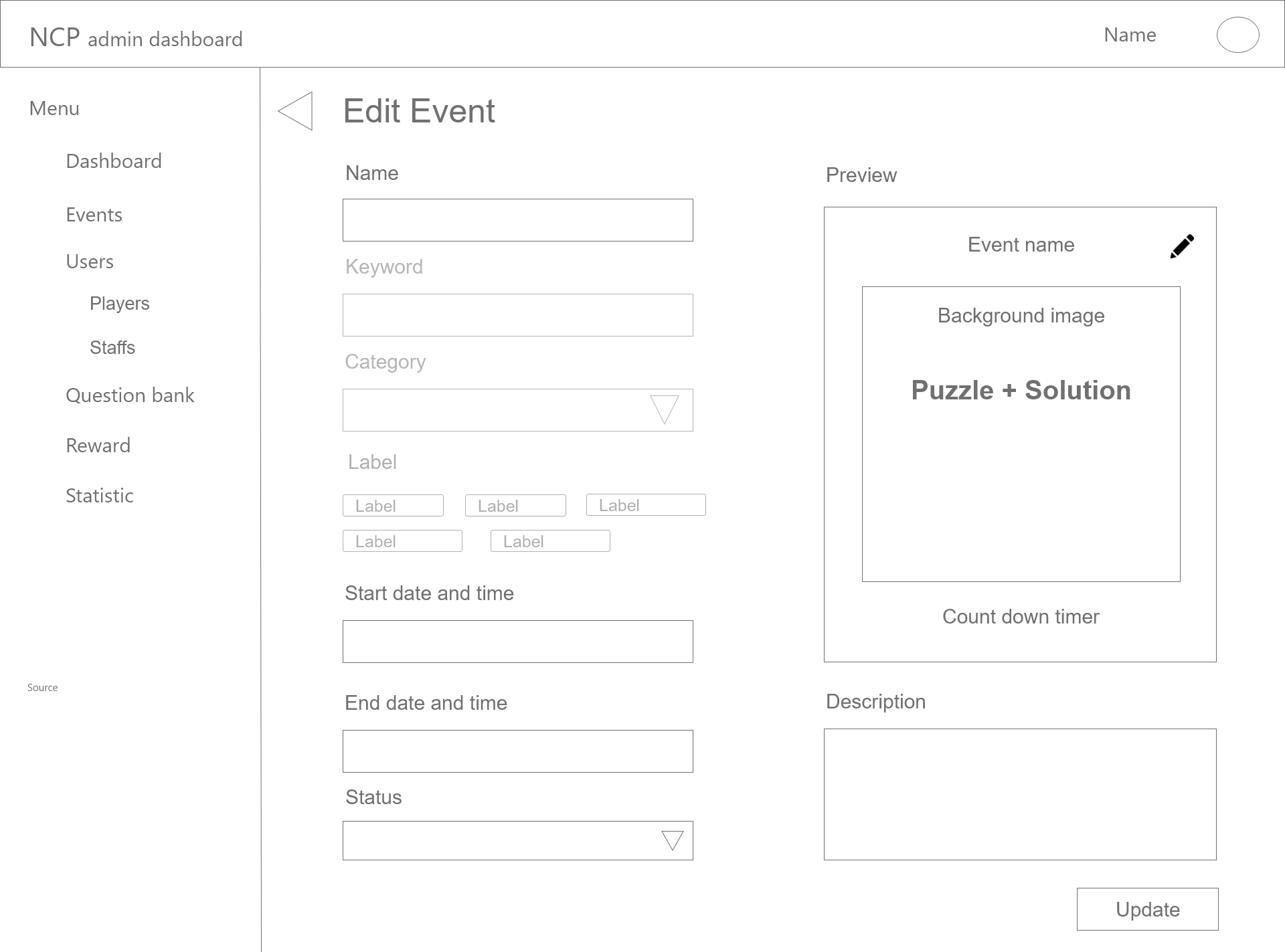
****

Figure 7 - Dashboard - Event - Upcoming event wireframe

The layout (Figure 7) shows the detailed information of the clicked event and it can be edited.

The main container of the layout consists of the information fields of the event(event’ name, keyword, category, labels, starting date and time, event description and status of the event: draft, published and canceled)) which can be edited except for the keyword, category and labels. On the top right corner of the main container of the layout, there is a preview box which can show the player perspective of that event that contains its information(event name, background image, puzzle and solutions, countdown timer), there is a button at the top right of the container that let the user change the background image. At the top corner of the preview panel, there is a button that lets the user change the background of the event.

On the bottom right corner, there is a button: Update. The Update button will confirm and save the information of the edited fields.

#### Create new event

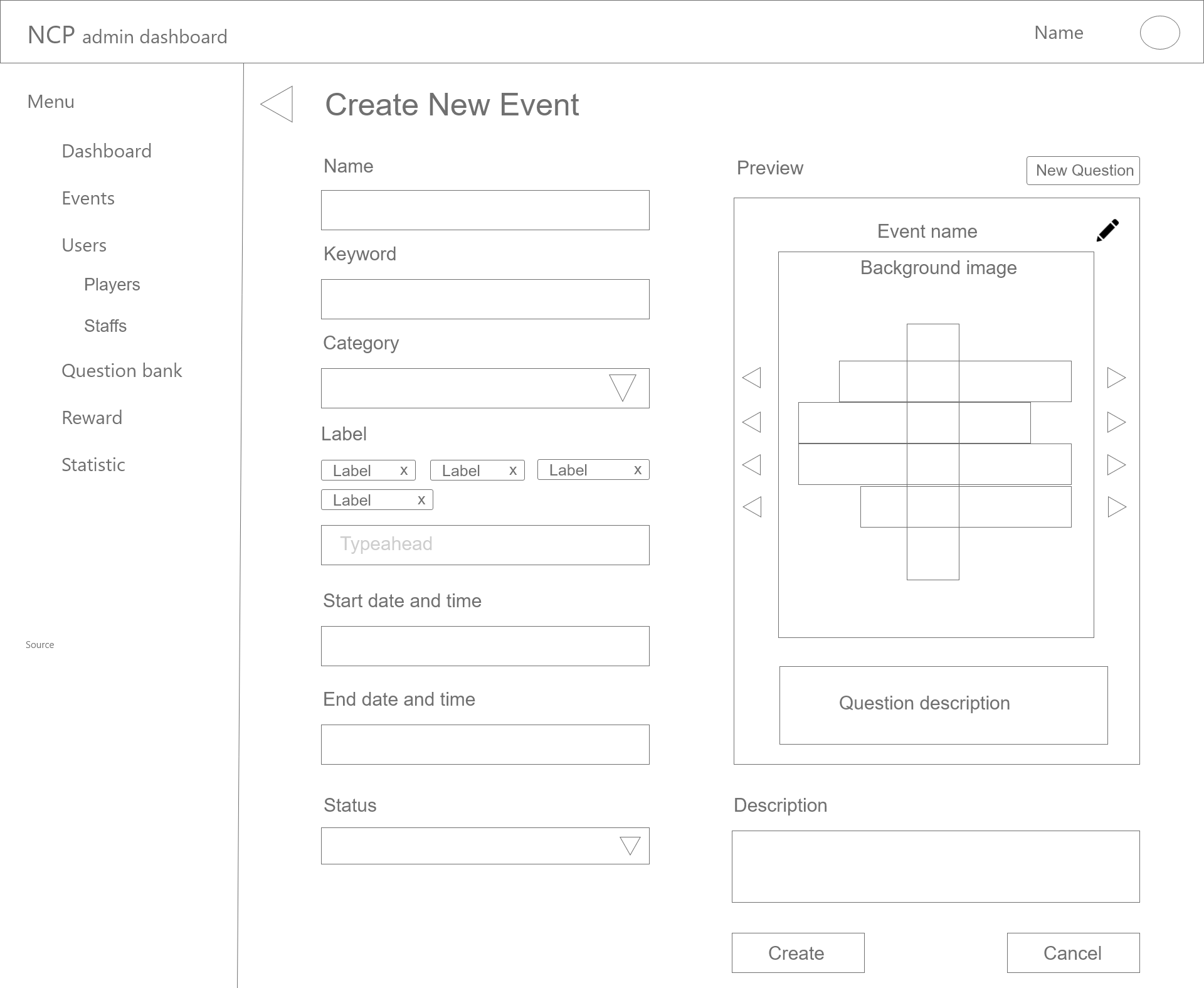
****

Figure 8 - Dashboard - Event - Create new event wireframe

The purpose of this layout (Figure 8) is to let the user create a new event for players.

The layout consists of all of the event information (name, keyword, category, labels, starting date and time, status). The right part of the layout is the preview of the event puzzle. The crossword puzzle is in the middle of the container with its keyword, there are previous and next buttons on both sides of the puzzle that helps the user change the keyword of that line.

There are two buttons at the bottom right corner of the layout. The first one is the Create button which helps the user to save and create the new event. The second one is the Cancel button which cancels the data and leads the user back to the previous layout (Figure 3).

The  button lets the user change the background of the event. And there is also a new question button on the top of the preview panel. Clicking on the new question button the layout will lead the user to the add question layout (Figure 10) (nice to have feature).

### Overview of the question bank

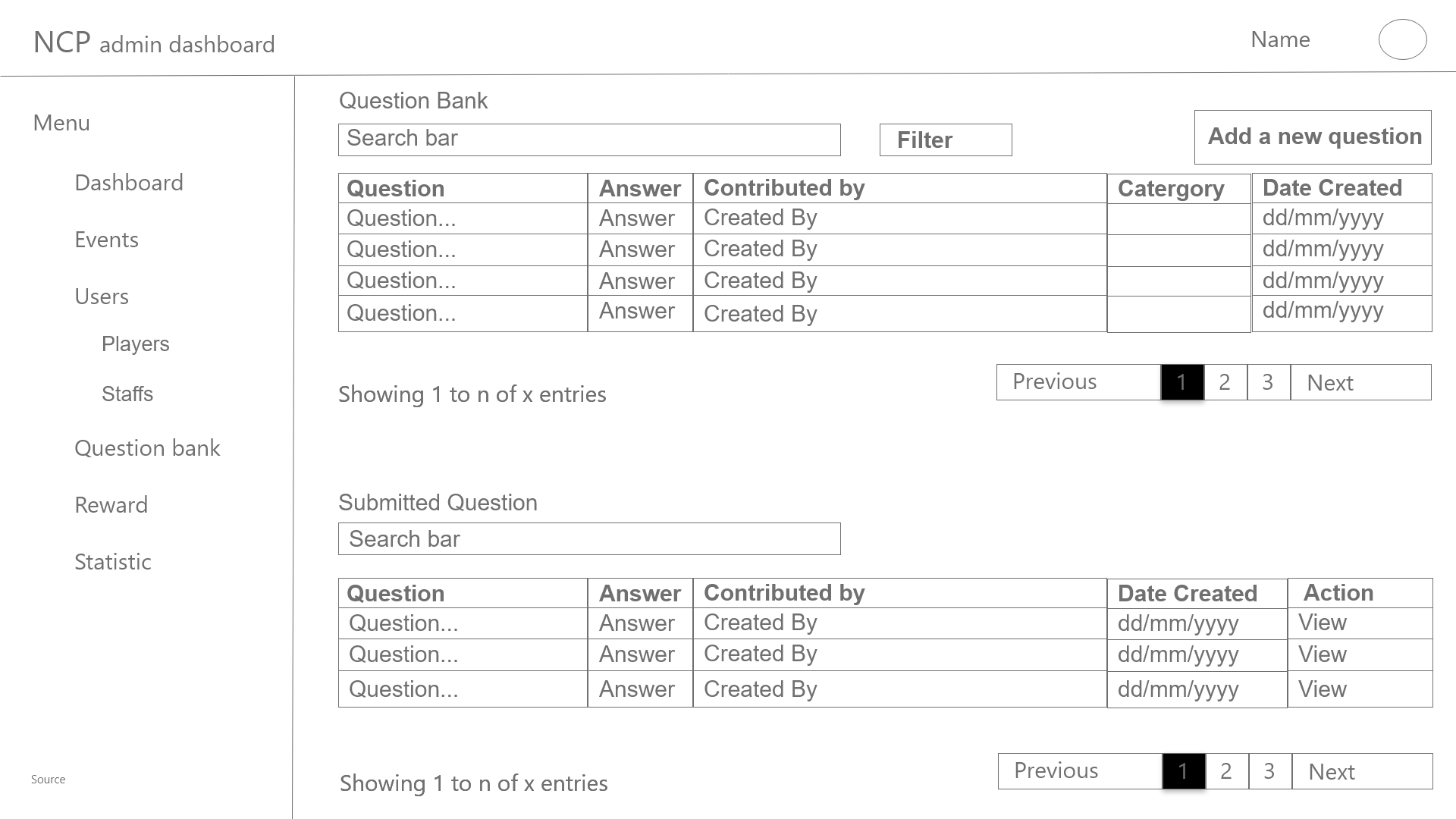
****

Figure 9 - Dashboard - Question bank wireframe

The display (Figure 9) shows the user the question bank of the system with their general information(. And also the submitted question part where display the submitted questions by the player that are waiting for the staff to approve. The display also provides an add new question option.

The question bank listed all the questions with their information(Question, Answer, Contributed By Who, Category and Date Created) which is on top of the container. This part also consists of a new question button.

The other part of the main container in the layout is the submitted question section. This section shows the submitted questions by the players by list with their information(ID, Question, Answer, Contributed by, Category, Date Created and action button)

All of the lists will have pagination.

#### Add a new question

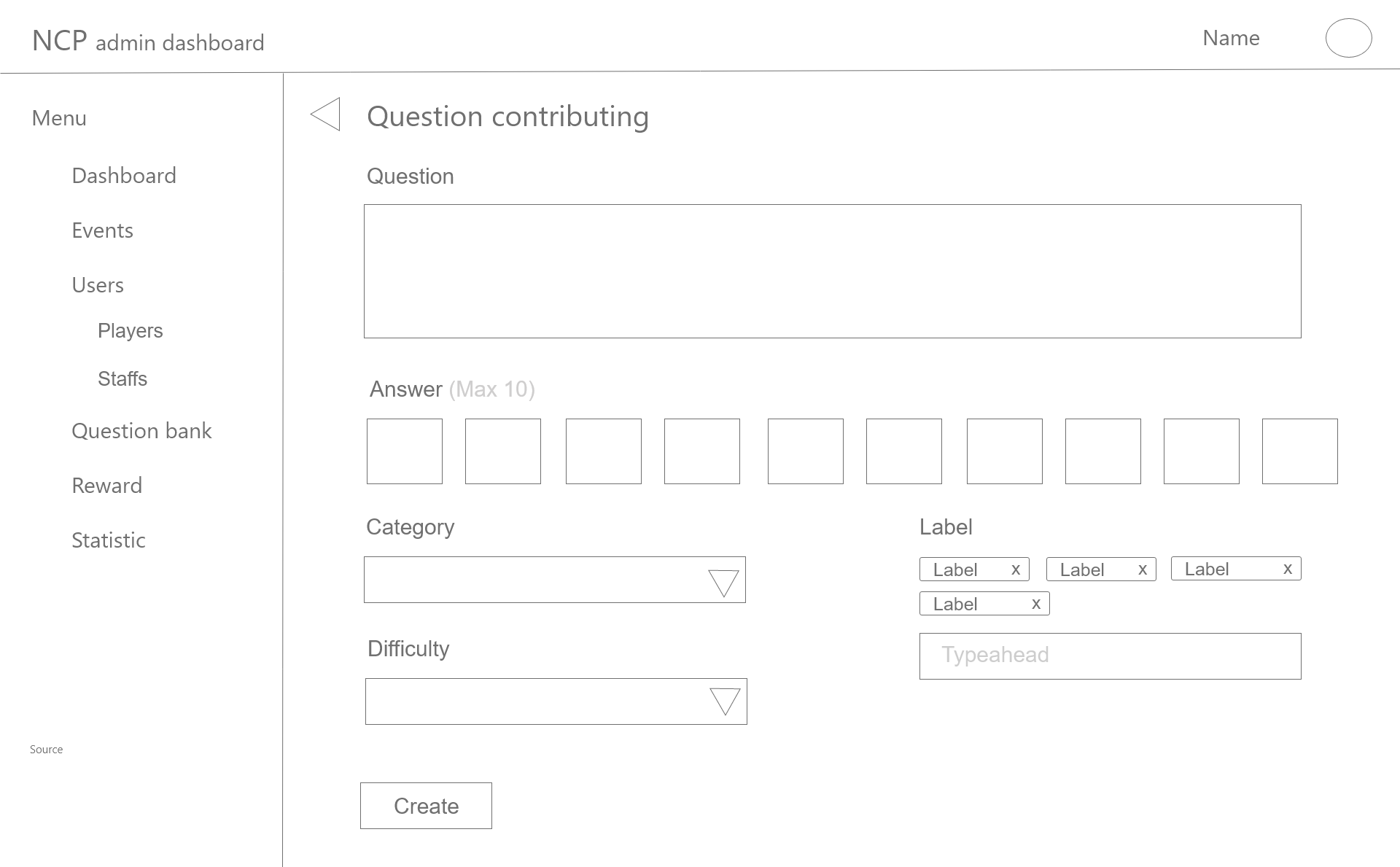
****

Figure 10 - Dashboard - Question Bank - Adding a new question wireframe

The purpose of this layout (Figure 10) is to let the current user (staff) add a new question into the question bank.

The main container of this layout consists of information fields of the question (question, answer, difficulty, category, label) that need to be filled in in order to add a new question to the question bank. The Create button lets the user add a new question if the user has filled in all fields and each field is verified. The difficulty (can only be set by HR staff), category and label, each of them have their own specific value that can be selected not by inputting in.

The triangle button on the top left corner of the main container of the display leads the user back to the previous layout.

#### View a question

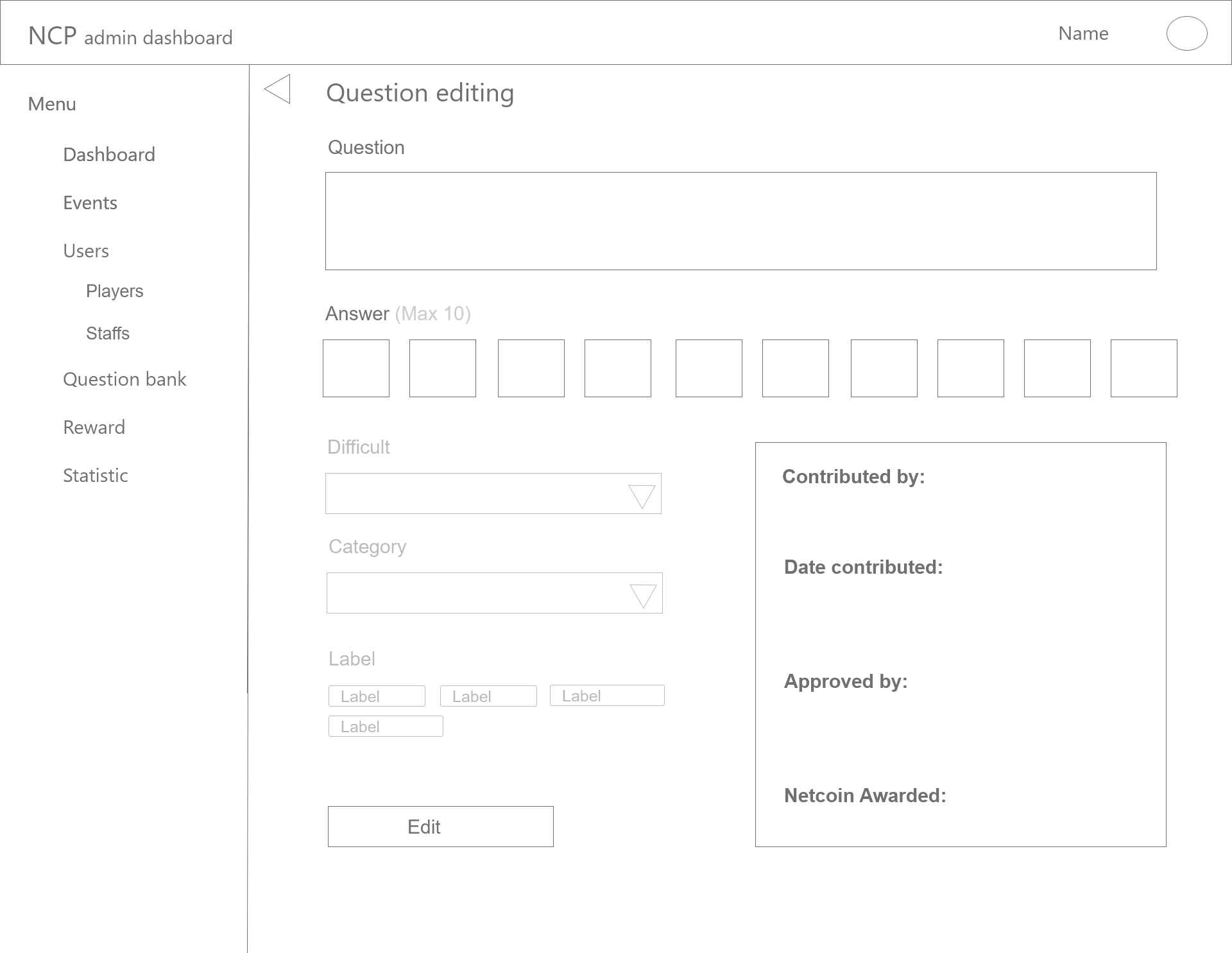


Figure 11 - Dashboard - Question Bank - View question wireframe

This layout (Figure 11) will appear after clicking on a specific question in the question bank layout (Figure 9). This layout shows the detailed version of the question, also the fields can be edited.

The layout contains all related information of the question ( (question, answer which is editable and also difficulty, category, labels, contributed by, date contributed, approved by and netcoin awarded fields which are not editable)

At the bottom right corner, there is an Edit button that lets the current user (staff) to save the edited fields which are changed.

The triangle button on the top left corner of the main container of the display leads the user back to the previous layout.

#### Approving a question



Figure 12 Dashboard - Question Bank - Approve question wireframe

This layout (Figure 12) will appear after clicking on a specific question in the submitted question list which is in the question bank layout (Figure 9). This layout shows the detailed version of the submitted question, also the fields can be edited.

The layout contains all related information of the question (question, answer, difficulty, category, label which is editable and also the contributed by, date contributed and status of the question fields which are not editable)

At the bottom right corner, there is an Approve button that lets the current user (staff) approve the contributed question and add it to the question bank, the Deny button will deny the question and delete it .

The triangle button on the top left corner of the main container of the display leads the user back to the previous layout.

### Overview of the user information

This section has two other small sections: Player overview and Staff overview.

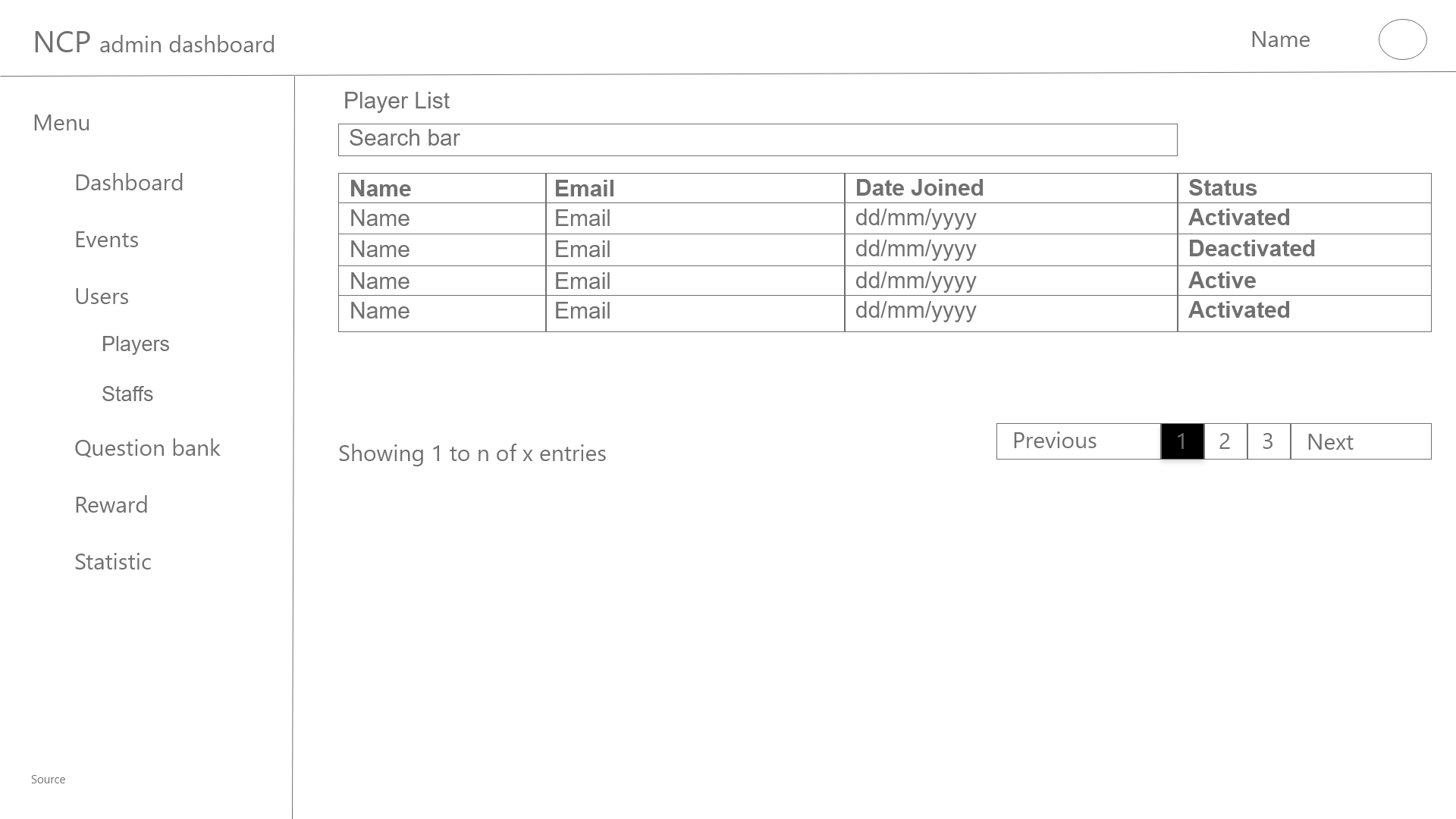
****

Figure 13.1 - Dashboard - Users - Player List wireframe

This layout (Figure 13.1) helps the user to quickly look through all of the players in the system in list with their general information. The layout consists of a search bar and sorting with each information column.

The list contains the general information of the user (name, email, date joined, status(activate, deactivate)). Clicking on each of the users will lead to the corresponding detailed profile page of that particular user. The list also will have pagination if it is too long.

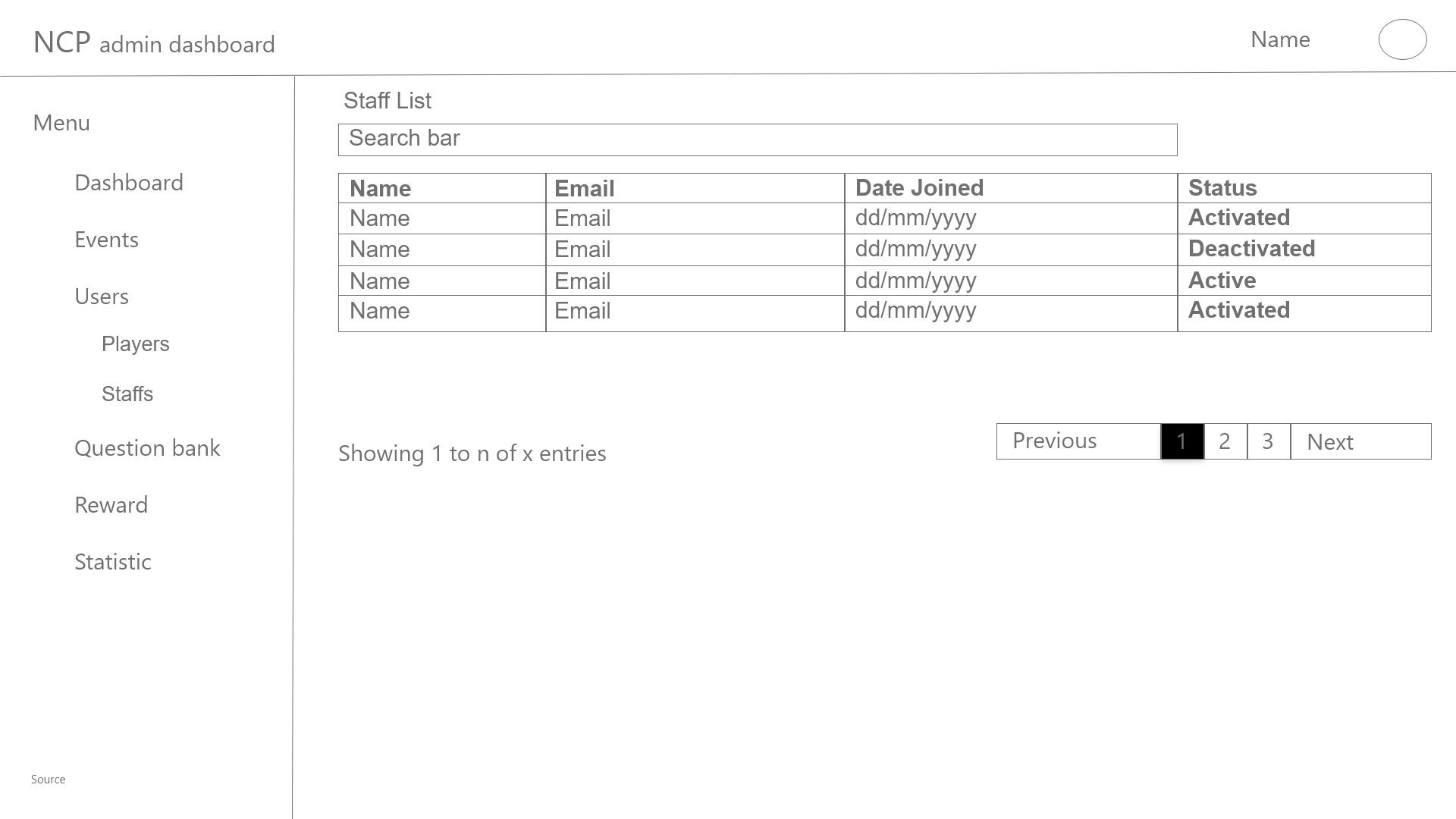


Figure 13.2 - Dashboard - Users - Staff List wireframe

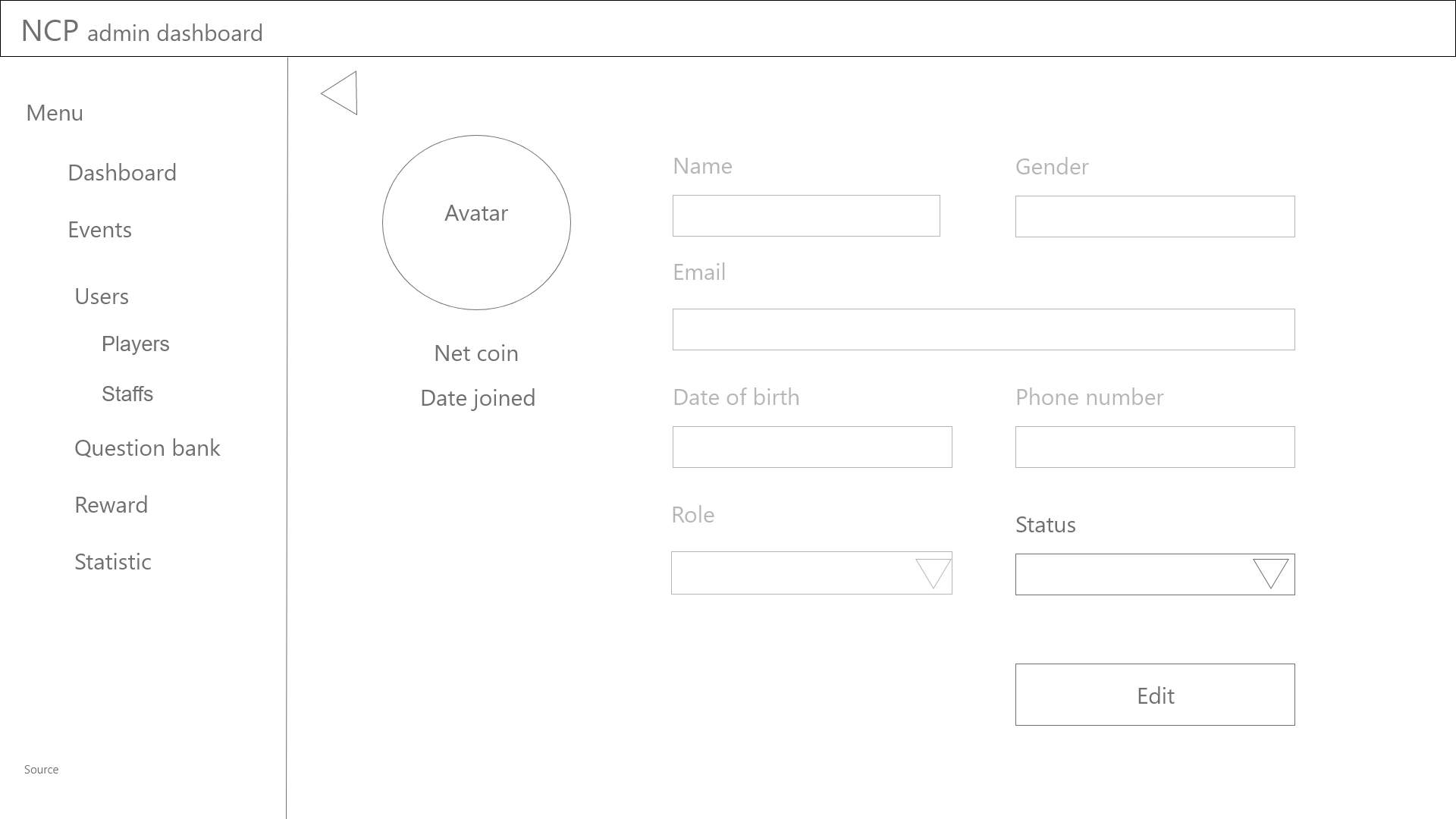
****

Figure 14 - Dashboard - Users - User profile wireframe

The main container of this layout (Figure 14) consists of information fields of the user(avatar picture, name, gender, date of birth, email, phone number, role(player, staff/admin once the role has been set to staff/admin, the role can not be set back to player and vice versa) and status(activate, deactivate)) which are editable. There are two buttons in this main container, the first one is the Back button which leads the user to the previous layout. The second button is the Edit button, which confirms and saves the edited information that the user has changed.

### Overview of the rewarding

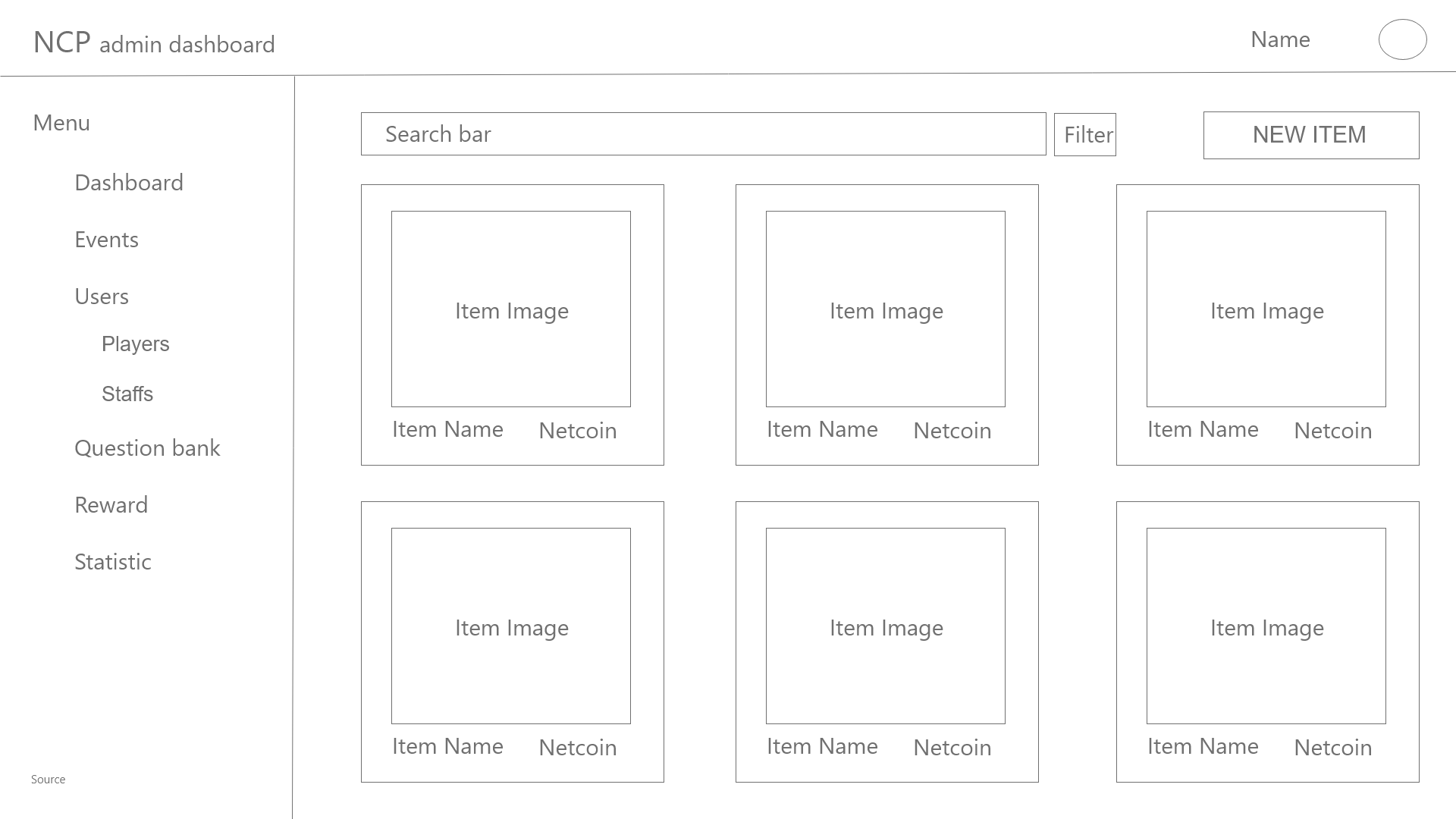


Figure 15 - Dashboard - Reward Manage wireframe

The purpose of this layout (Figure 15) is to let the user see the list of all items of the exchanging system with general information (Name, photo, Netcoin cost).

The main container of this layout contains many small containers which are items. Each of the small containers is included with the item photos, item name and item cost. There is also a search bar and a filter tab. Clicked on each item will lead to the detailed information page of that item.

The button at the top right corner is the add new item button.

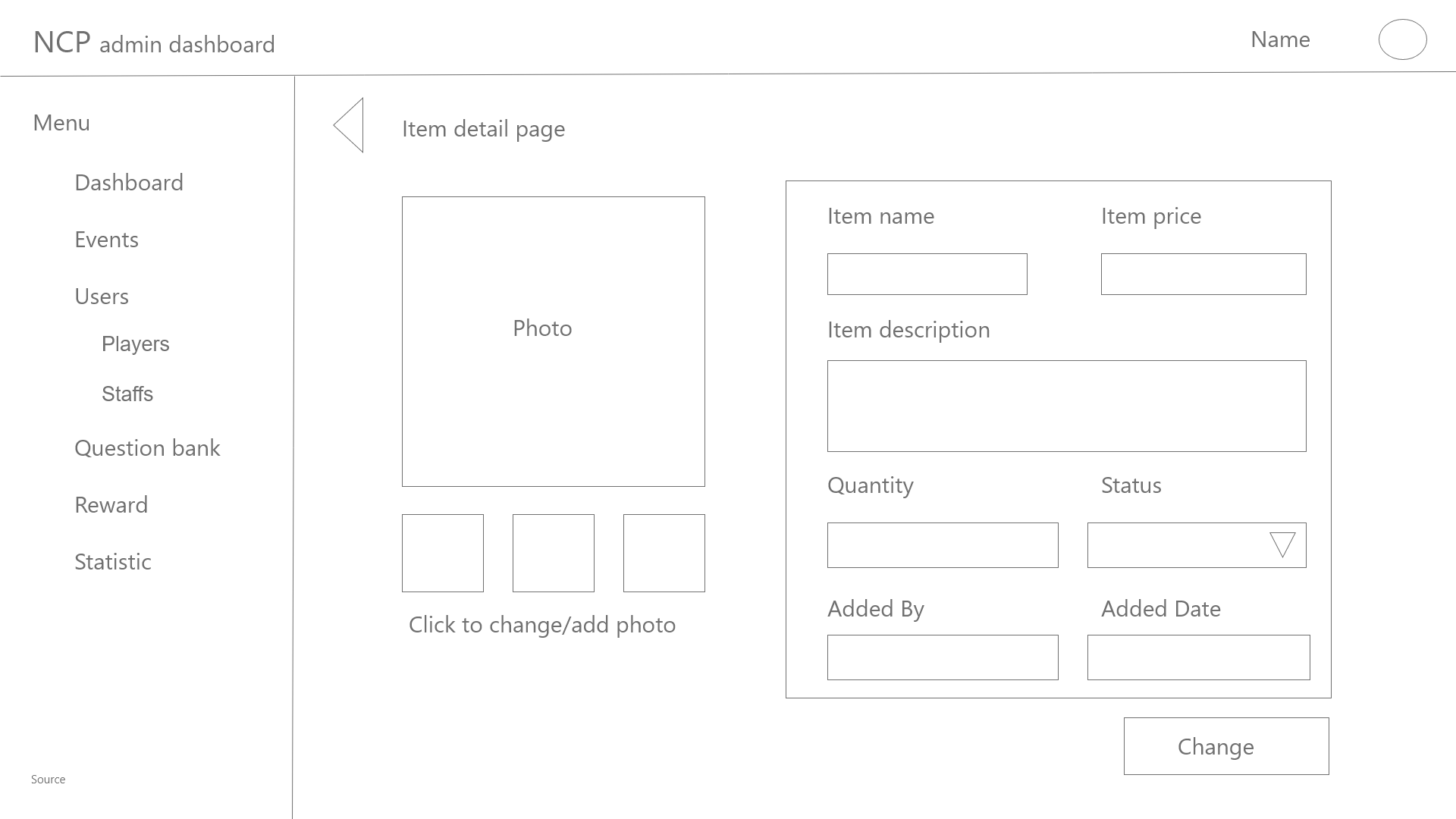


Figure 16 - Dashboard - Reward Manage - Item detailed wireframe

After clicking at an item in the Reward Manage layout (Figure 15), this layout (Figure 16) appears. The purpose of this layout is giving the user a detailed view of the corresponding item.

The layout consists of the item information fields (name, price, description, quantity, status, added by, added date) which are editable. The left part of the layout is about the item photos, which can be added/changed or deleted.

There are two buttons in this layout, the first one is the Triangle button which leads the user back to the previous layout and the other one is the Change button which lets the system save the edited information fields that have been changed by the user.

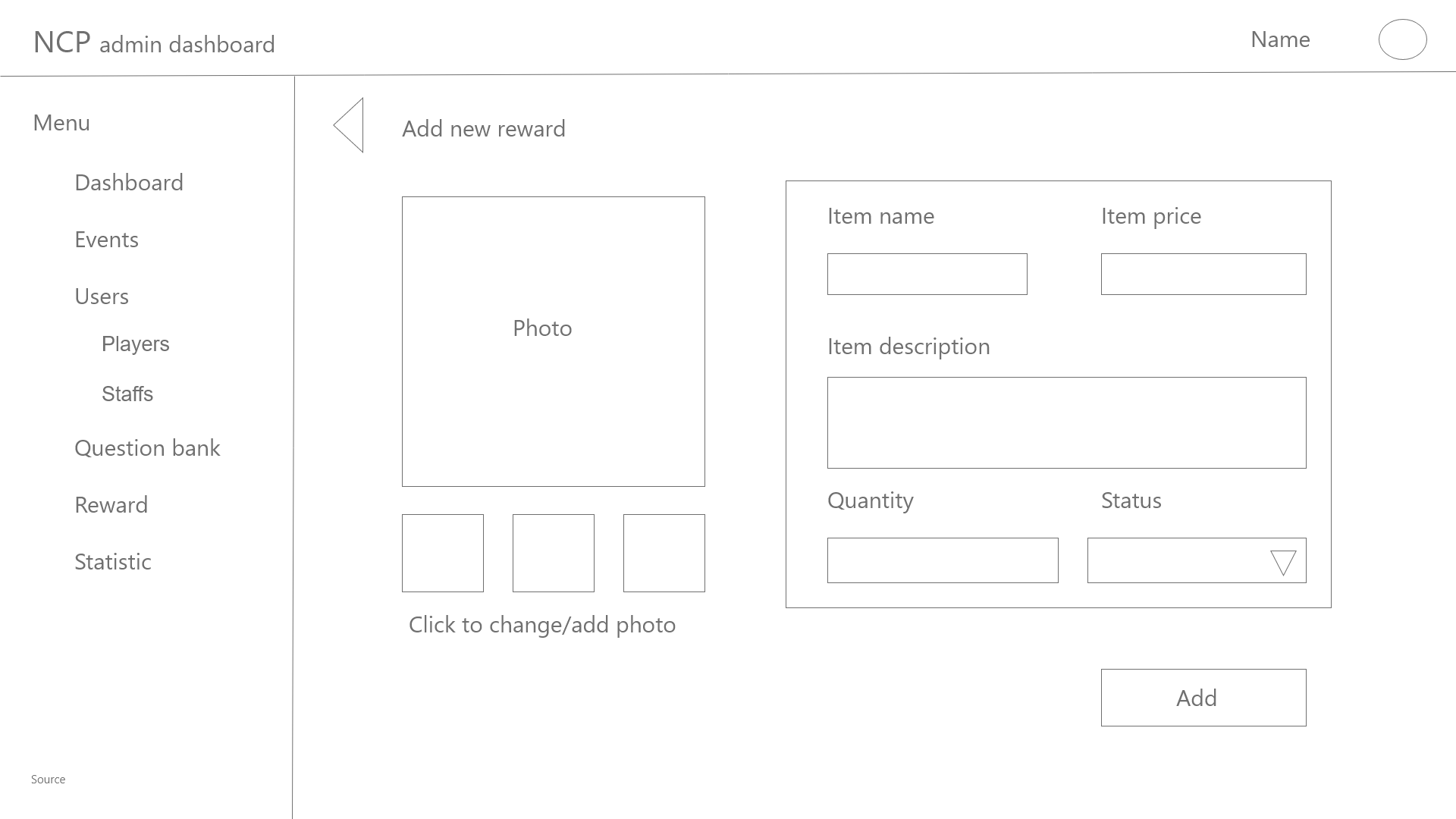


Figure 17 - Dashboard - Reward Manage - Add new reward wireframe

The layout (Figure 17) lets the user add a new exchanging item to the system. The layout consists of the item information fields (name, price, description, amount, status) which are editable. The left part of the layout is about the item photos, which can be added/changed or deleted. The left part of the layout is about the item photos, which can be added/changed or deleted.

There are two buttons in this layout, the first one is the Triangle button which leads the user back to the previous layout and the other one is the Change button which lets the system save the edited information fields that have been changed by the user.

## Authentication / Authorization

### Authentication / Authorization

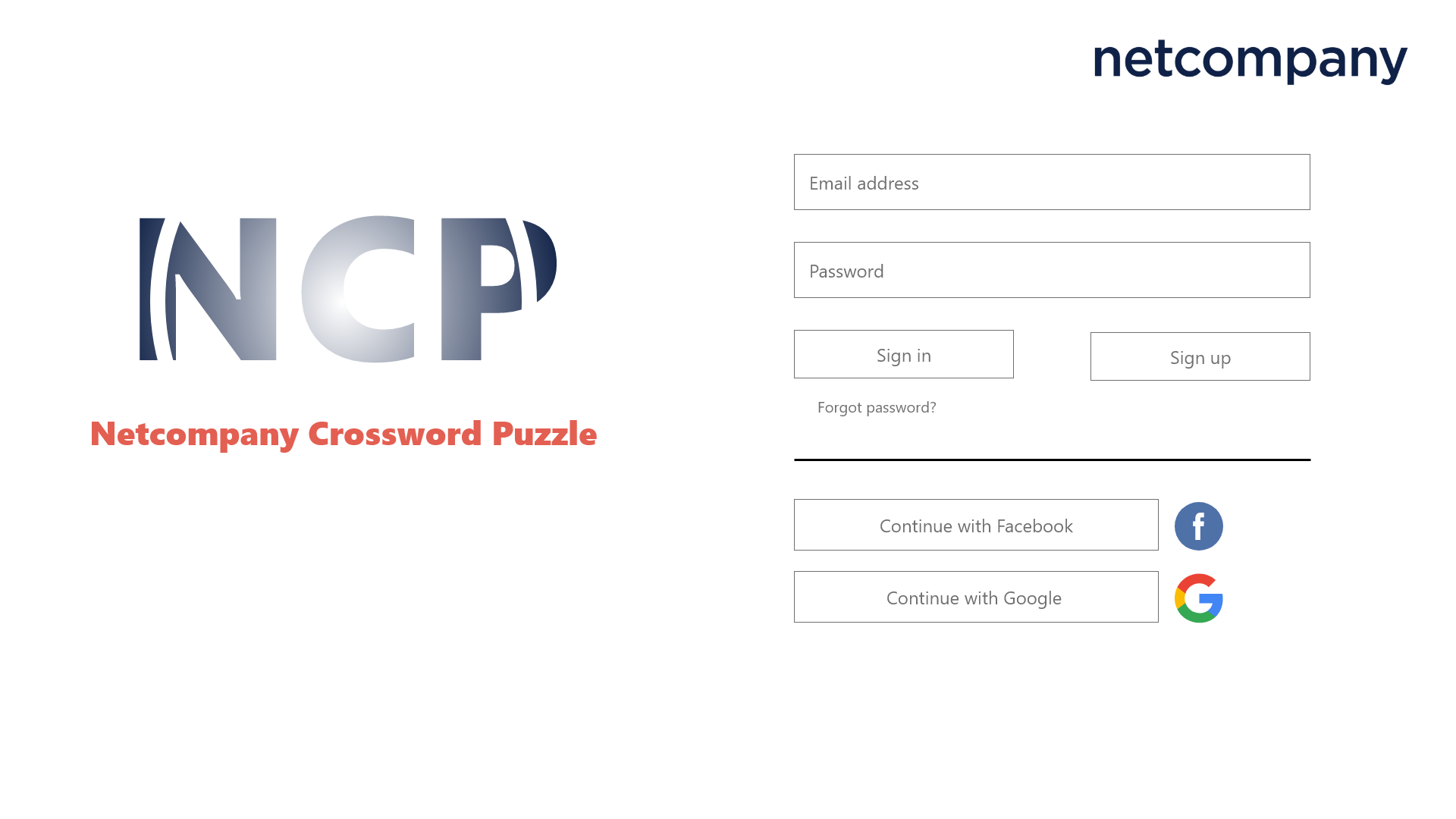
This authentication page is used to provide authentication for the user to login to their account, the detailed layout is shown in the figure below.

Figure 18 - Authentication / Authorization - Authentication

In this figure, there is the netcompany logo in the top right corner in the heading area.

In the main container area, it is divided into 2 parts, on the left side is the main logo of the Netcompany Crossword Puzzle project. On the right side consists input fields and option buttons for the user to choose for authentication or authorization.

First there is input fields of the user’s account information (email, password) that users need to fill to login to their account

Next, there are 2 buttons, the Sign in button on the left side and the Sign up button on the right side. The sign in button when activated will log the user in to the account based on information on the input fields. The sign up button when activated will create a new account with basic information taken from the input fields and the user can fill more or change later (see Change information section for more information).

Next is the Forgot Password field, when activated it redirects to the Forgot password layout (see Forgot password section for more information).

Last is 2 buttons, the Continue with Facebook button and Continue with Google button. Those buttons require the user’s authorization to create a new account and synchronize their information from their Facebook/Google account to this new account.

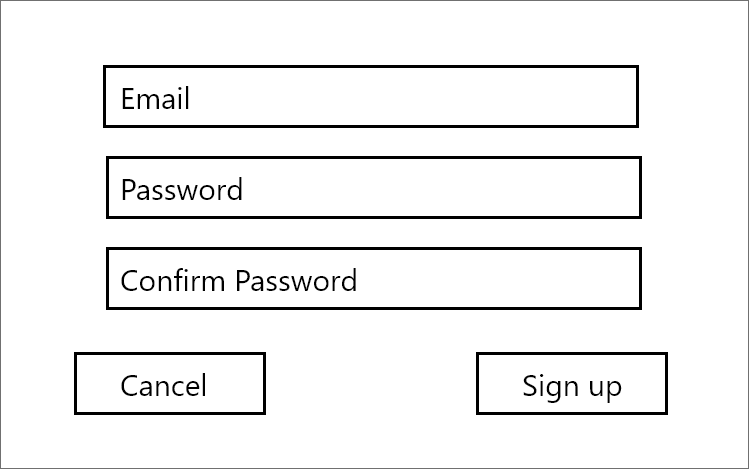


Figure 18-1 -Authentication / Authorization - Confirm sign up

The main container of this figure including 3 input fields respectively are for email, password and confirm password

Last is the 2 buttons, on the left side is the cancel button, when clicked it will return the user to the previous page, on the right the right side is the sign up button when clicked it creates a new account with the above information in the database and log the user into the dashboard after that.

### Fail to sign in

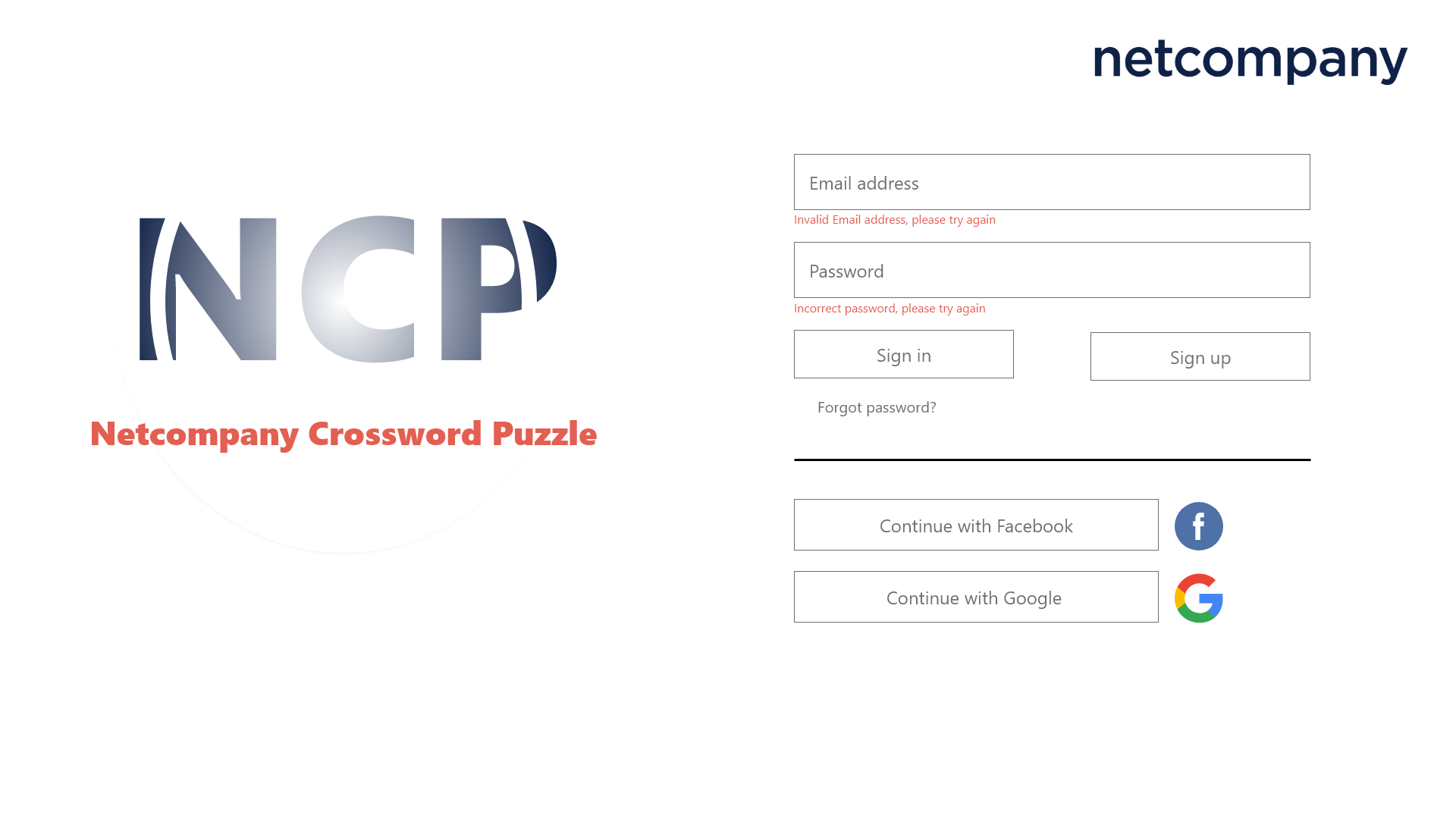


Figure 19 - Authentication / Authorization - Fail to sign in

This layout appears when the information that the user fills in figure 18 is not matching with the information in the database and cannot log the user in.

The basic containers will be the same as figure 18.

Red lines will appear under the input field where the filled information is wrong to announce the error.

### Forgot your password



Figure 20 - Authentication / Authorization - Forgot your password

This figure appears when the user clicks on the Forgot your password in figure 18 and figure 19.

The main container includes 2 input fields, one is the Email address that the user needs to fill to receive a reset password code. The other input field is for the user to import the verification code to reset password.

Next is the send again which is above the reset password code input field, and is used to resend another code to the email address in the Email input field.

Last are 3 buttons, first is the send button located above, used to generate and send a verification code to the email address in the Email input field. Second is the cancel button located below on the left, used to cancel the reset password process and back to the figure 18 layout. Third is the Submit button, used to verify the code and redirect to the reset password layout (see reset password section for more information).

### Verification code incorrect



Figure 21 - Authentication / Authorization - Verification code incorrect

The main container of this figure is the same as figure 20. When the verification code which is sent to the email address of the user but the user fills it wrong in the reset password code input fill field, the warning red line will appear under the reset password code input fill field.

### Reset Password

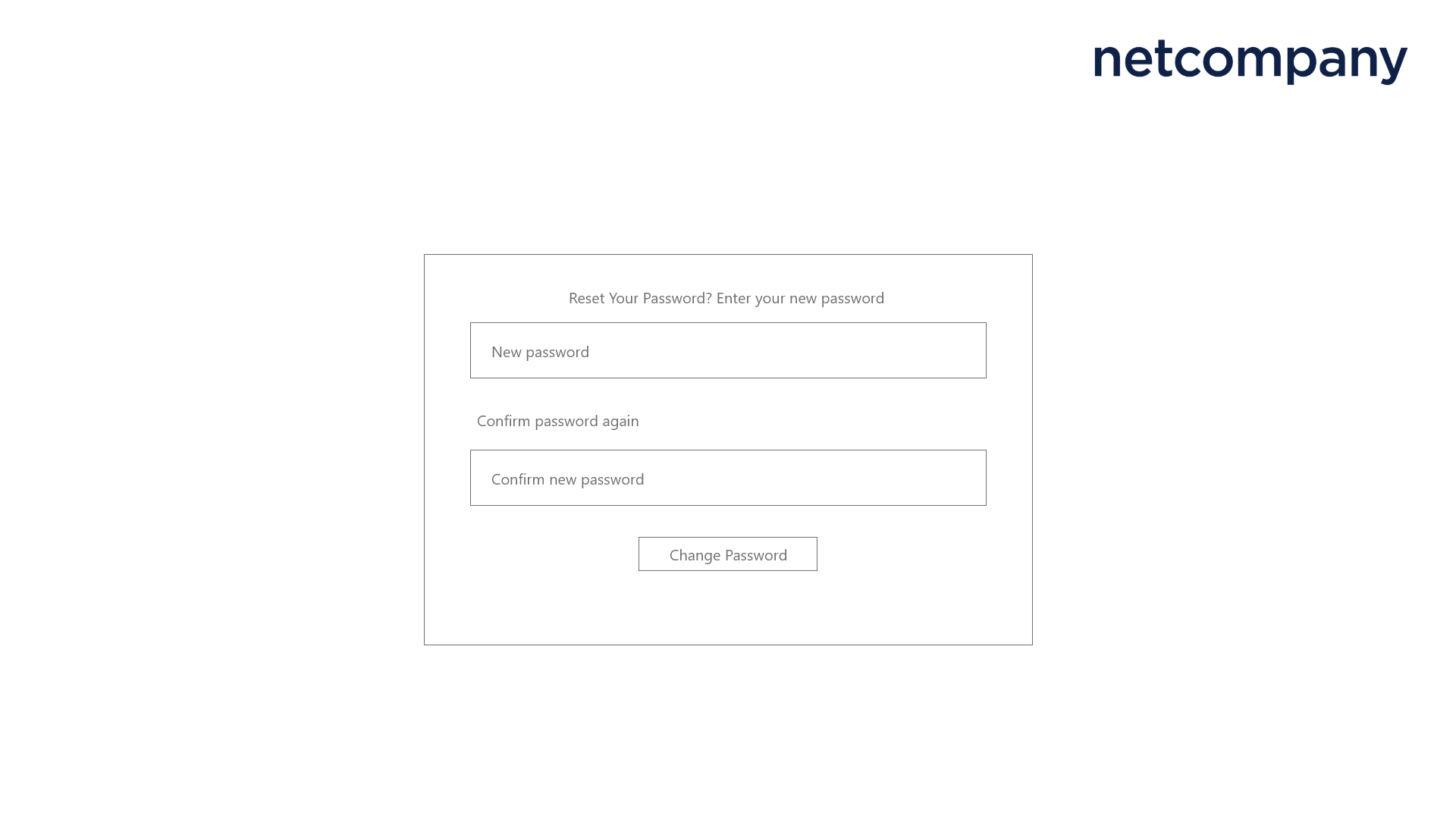


Figure 22 - Authentication / Authorization - Reset Password

This layout appears when the verification code in the figure 20 is confirmed and is used the reset the password of the account connected to the email address in figure 20

There are 2 input fields, the first one above is the new password input field, the user needs to fill the new password other than the old password. The second one below is the confirm new password, the user needs to re-input the new password to confirm there is no mistake at the new password.

Last is a Change password button, used to confirm and change the old password to the new password.

### Confirm reset password unmatch



Figure 23 - Authentication / Authorization - Confirm reset password unmatch

The main container of this figure is the same as figure 21. When the user fills the confirm new password input field and it is unmatched with the new password input field , the warning red line will appear under the confirm reset password code input field.

## Dashboard (Player)

### Main dashboard

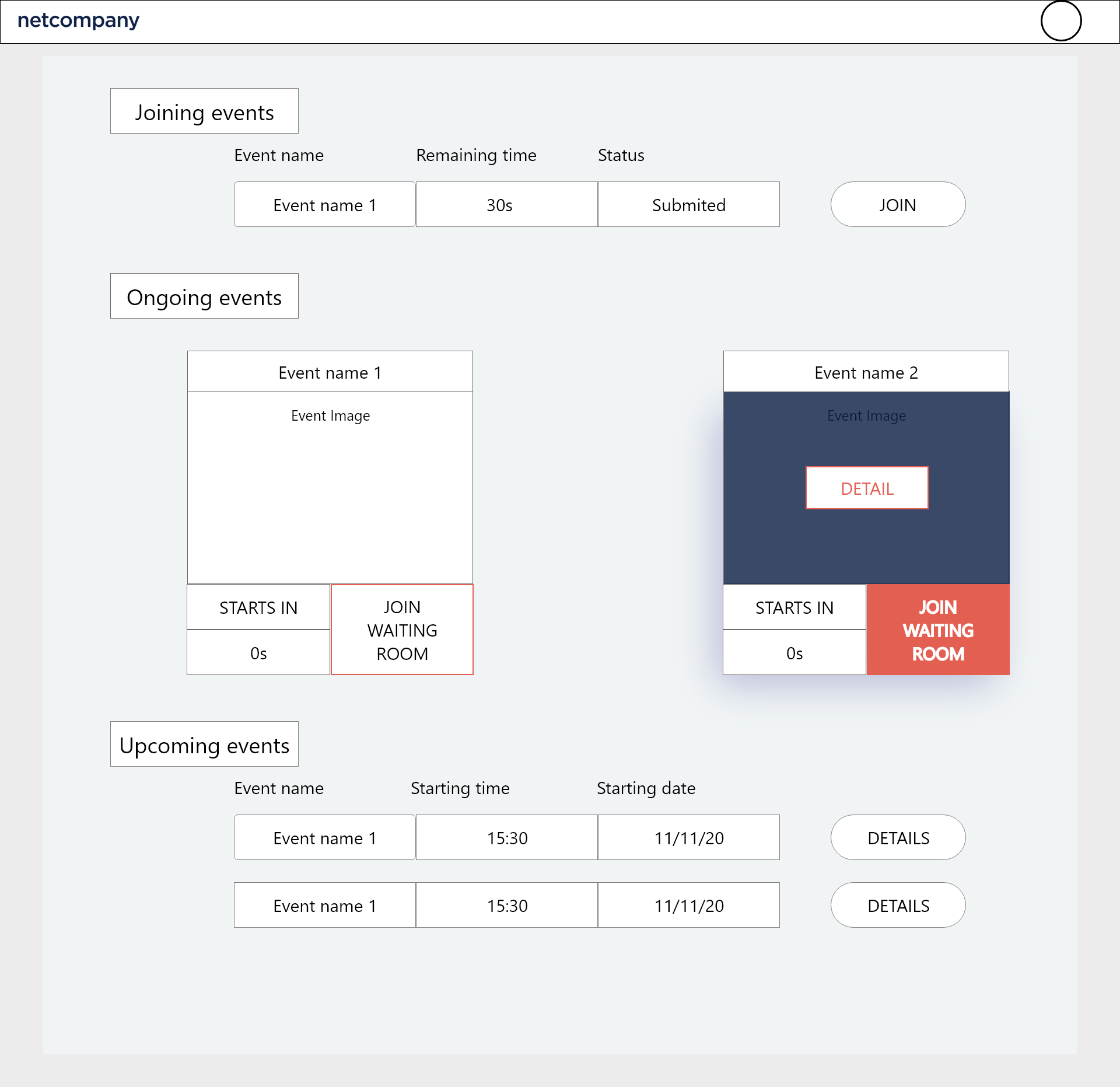


Figure 24 - Dashboard

This layout is when the user logs into the account successfully. The purpose of this is to work as a main page of a user (player). It will display the name of the event, the time remaining and the current status of that event (player submitted or not submitted).

The dashboard has multiple containers, including heading, main container consists of the Joining events of that player, the ongoing events area and upcoming events area.

The heading will have the avatar of the user. By clicking on the avatar, the user menu will be displayed (see figure 28).

The main container area will work as a scrollable area and contains joining events area, ongoing events area and upcoming events area.

In the joining events area, it will display events that the player already joined but leave for tended or untended reasons.

In the ongoing events area, it will display events in the form of an image including name, detail button and time countdown to the time when it starts and join the waiting room button.

The event image has a hover, when the user moves the mouse to that area, the hover with blue background and with a detail button in it will show up. when clicked, the detail button redirects to the event detailed figure.

The join the waiting room has a hover, when the user moves the mouse to the area, the hover with red background and white text will show up. When clicked, the join the waiting room button redirects to the waiting room figure.

In the Upcoming events area, it will display events in the form of a table with Details buttons, these details button allows the user to see the detailed information of events in the table .Events get moved from Upcoming events to Ongoing events when the start time <= 15 mins.

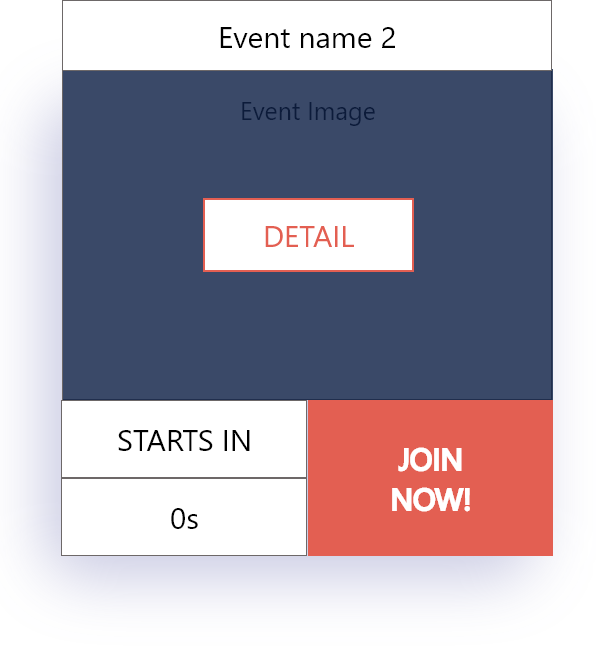
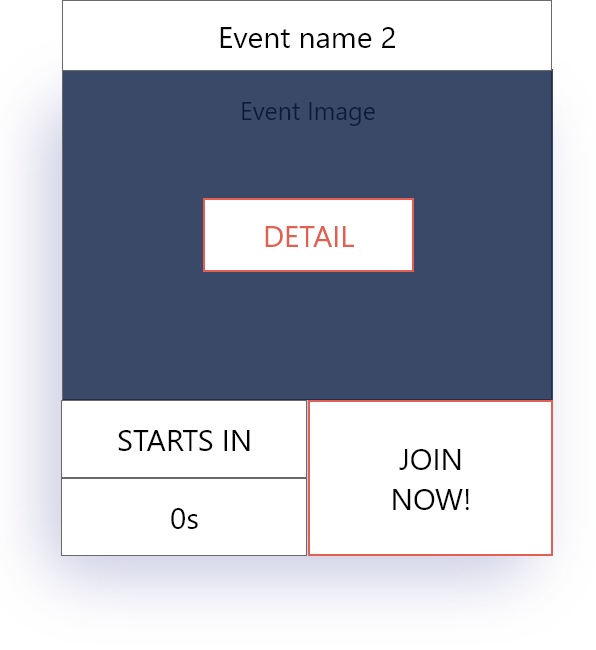


Figure 25 - Dashboard - Timer hits 0

In figure 24, when the countdown timer of an event hit 0, the join the waiting room button will change to the join now button. The join now button has a hover, when the user moves the mouse to that area, the hover will show up with a red background and white text. When clicked, the button redirects to the playing figure.

### Event details

In figure 24, when the user clicks on the detail button of a happing event, it redirect to this figure

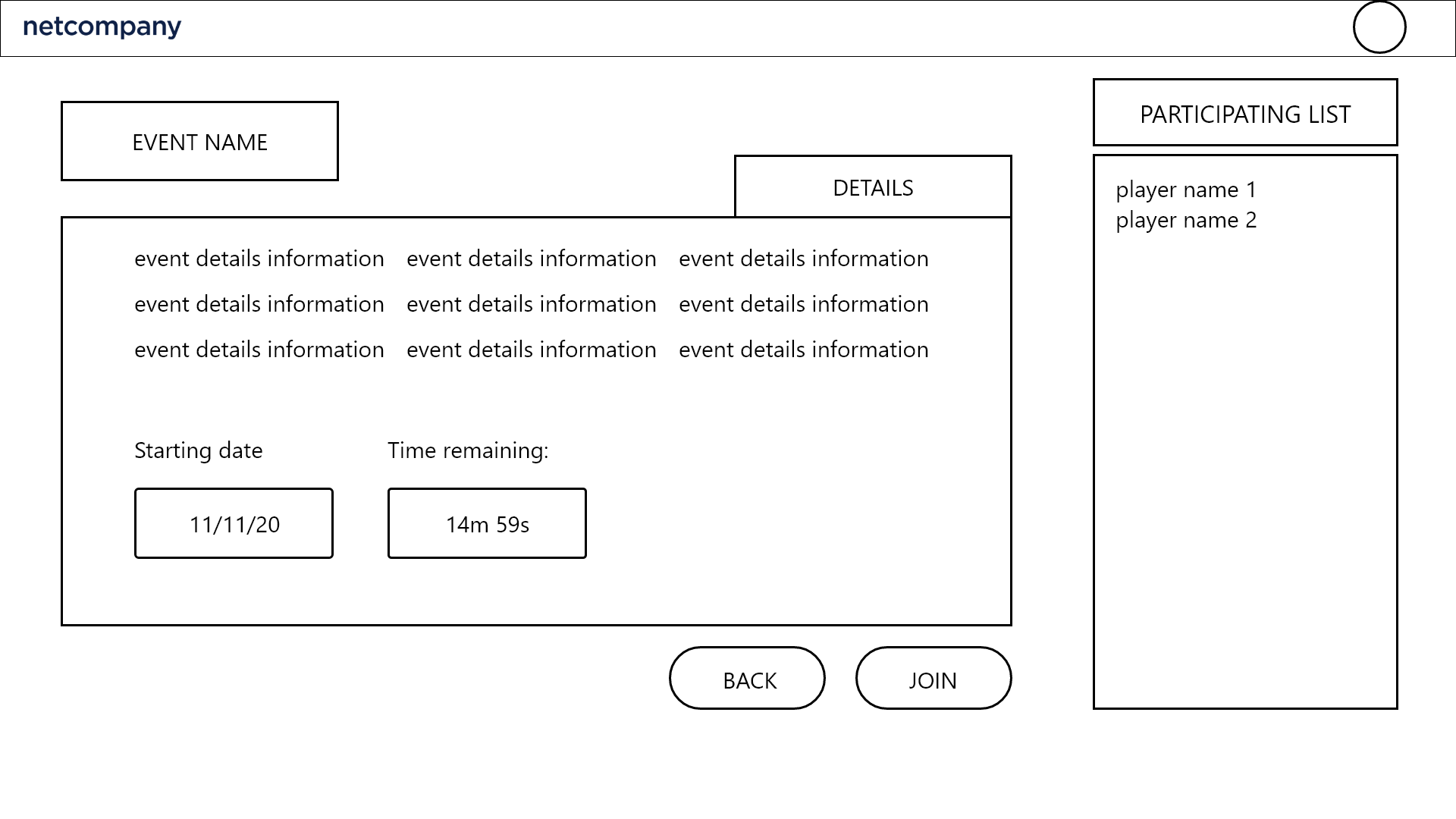


Figure 26 - Dashboard - ongoing event information

In the main container of this figure, on the left side is the Information of the event, including the name of the event, detailed information about that event, who is the host, the starting date and time remaining until it starts.

On the right side is the participants list, it shows names of users who already join the waiting room.

Last is the 2 buttons, on the left side is the cancel button, when clicked it will return the user to the previous page, on the right the right side is the join button when clicked it joins the user to the waiting room (see figure 31).

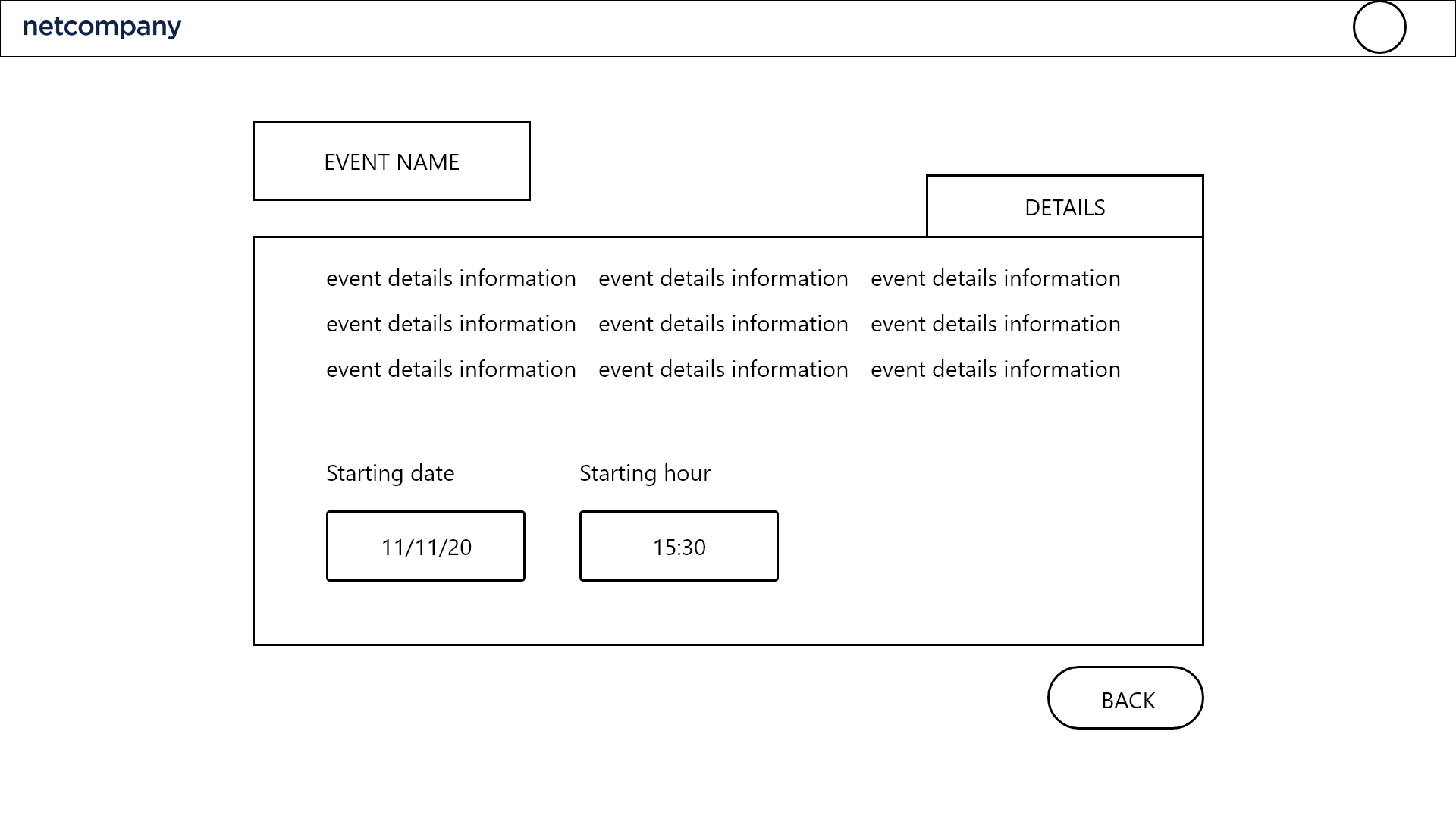
In figure 24, when the user clicks on the detail button of a upcoming event, it redirect to this figure

Figure 27 - Dashboard - Upcoming event information

In the main container of this figure, there is the Information of the event, including the name of the event, detailed information about that event, who is the host, the starting date and time when it starts.

Last is the back button, when clicked it will return the user to the previous page.

### User menu



Figure 28 - Dashboard - User menu

This figure will appear as an overlay whenever the user clicks on the avatar in any other figure which has heading with the user avatar and located right below the avatar.

The figure consists of the user's basic information, including player’s rank in total and current netcoin having and has 5 buttons including Matching history, Shop, Change information, Contributing and Logout, which when clicked will activate and redirect to their section figure.

## Contributing (Player)

In the figure 28, when click on the contribution button, it will redirect to this contributing layout figure

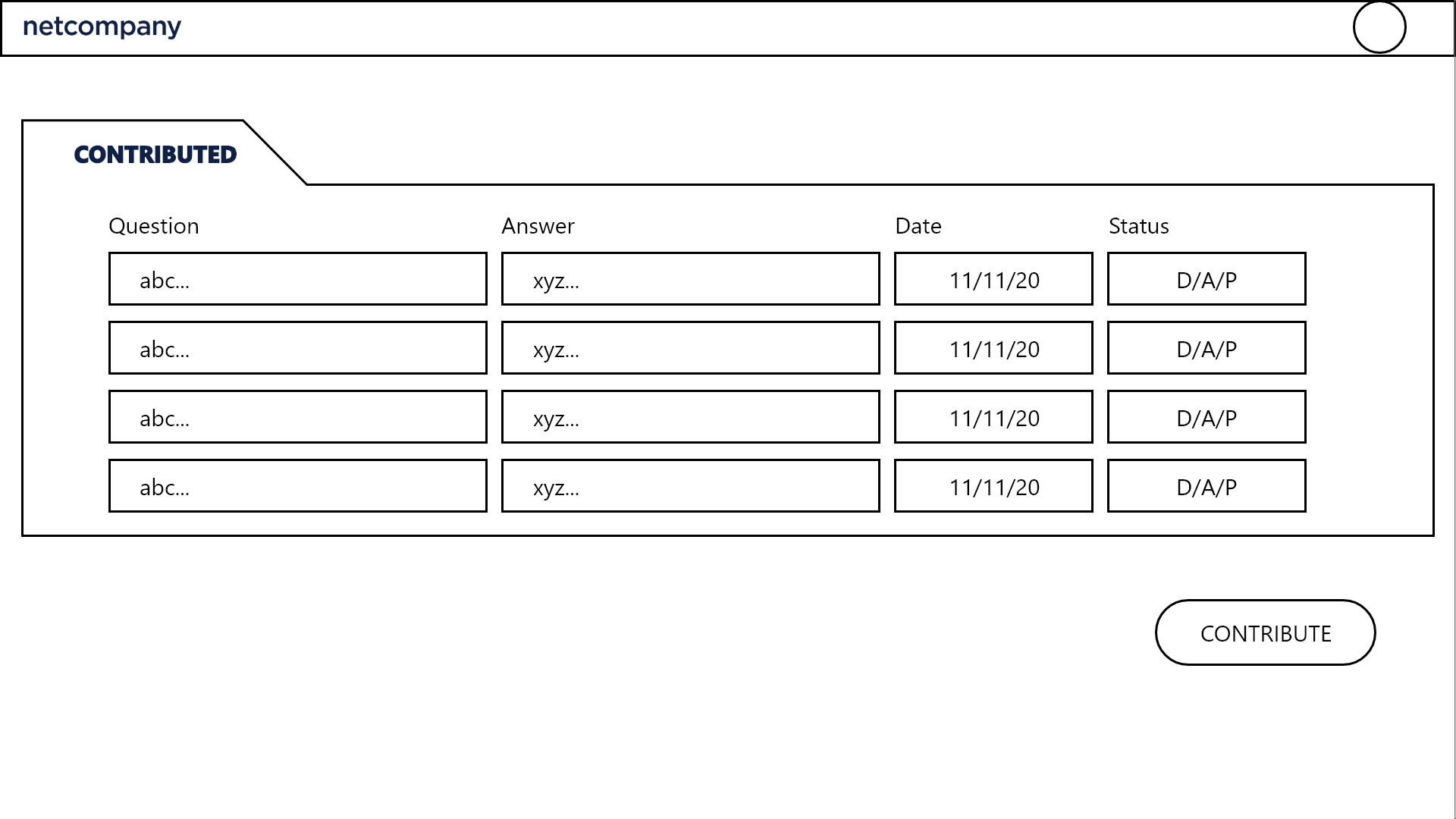


Figure 29 - Contributing - Contributed questions

In the main container of this figure, the contributed question is in the form of a table, including the question, the answer, the date contributed and the status of that question (accepted, denied, pending). Below the contributed question is the contribute button, when clicked it pop-up the contribute form for the user to contribute a new question (figure 25).



Figure 30 - Contributing - Contribution

This figure appears as a pop-up from figure 24. In the top right corner is the question contributed area, it shows how many questions the user contributed on that day and in that month.

Below are the input fields of the question and the answer. When the user types the answer in the answer input field (maximum 10 letters), the answer review area will auto generate the number of boxes equal to the number of letters of the answer (maximum 10 letters) and then auto fill the boxes with the letters of the answer.

Last are the 2 buttons, on the left side is the cancel button, when clicked it will return the user to the previous page, on the right the right side is the exchange button when clicked it sends the contributing question form to the admin to decide if that question is accepted or denied.

## Rewarding (Player)

In figure 24 and figure 33, if a user's question is accepted or the user is in the top 3 player, the user will be rewarded netcoin (see contributing section and gameplay section for more information).

## Exchanging system (Player)

In the figure 23, when click on the Netcoin shop button, it will redirect to this Shop layout figure

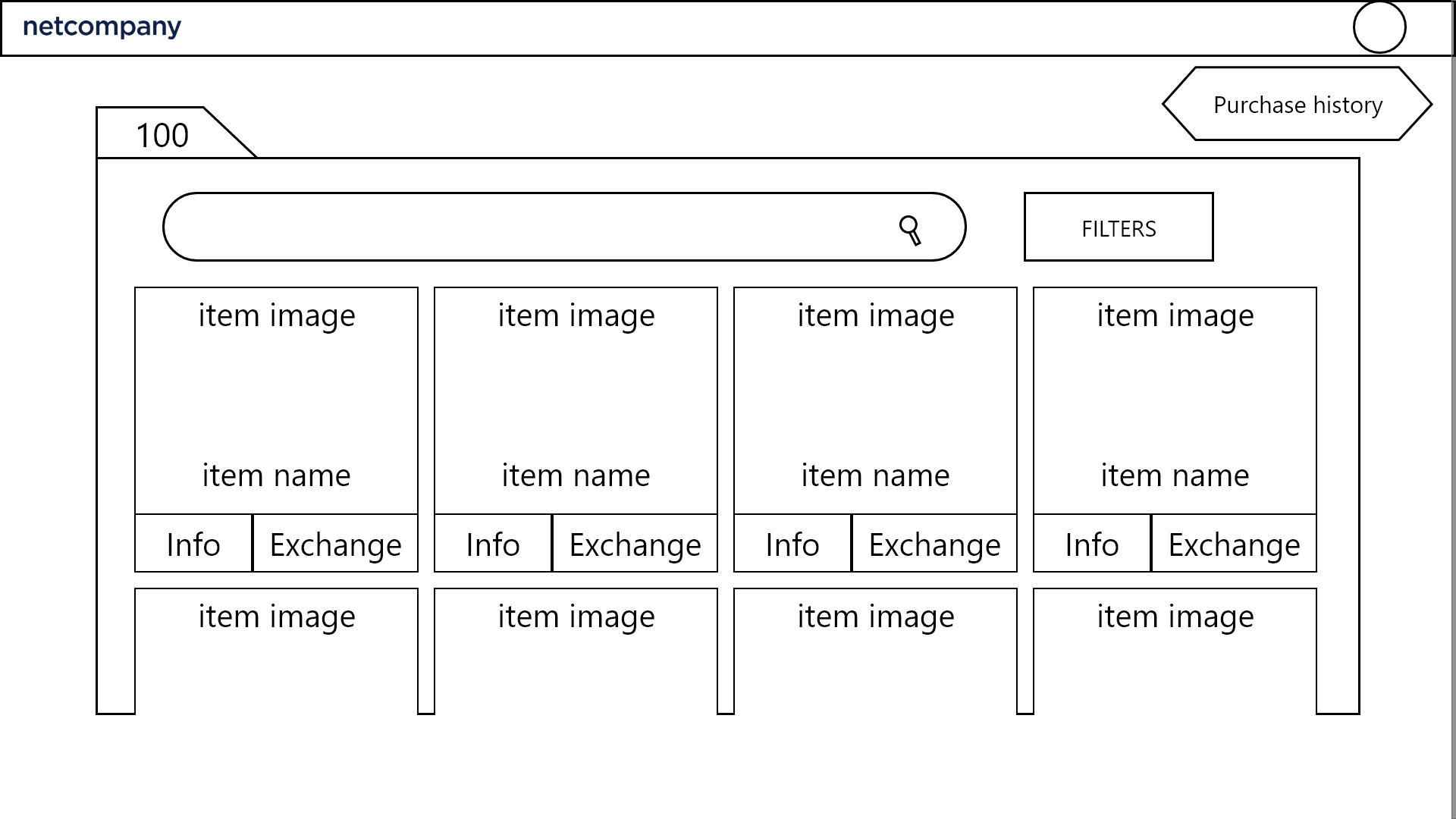


Figure 31 - Exchanging system - Netcoin shop

In the main container, at top right corner is the purchase history button, when clicked, it pop-up the history of items that the user purchased in the form of a list, including name, date of purchase, quantity and price.

On the top left corner is a number which presents the amount of netcoin the user is having.

Next is the search bar and filter button, the filter button will sort items based on price (high to low, low to high) and alphabet.

The list of item will be displayed in a grid view with each item is displayed in the form of a image with 2 buttons, the information button on the left side and exchange button on the right side

The information button when clicked will pop up the detailed information of the item (see figure 32).

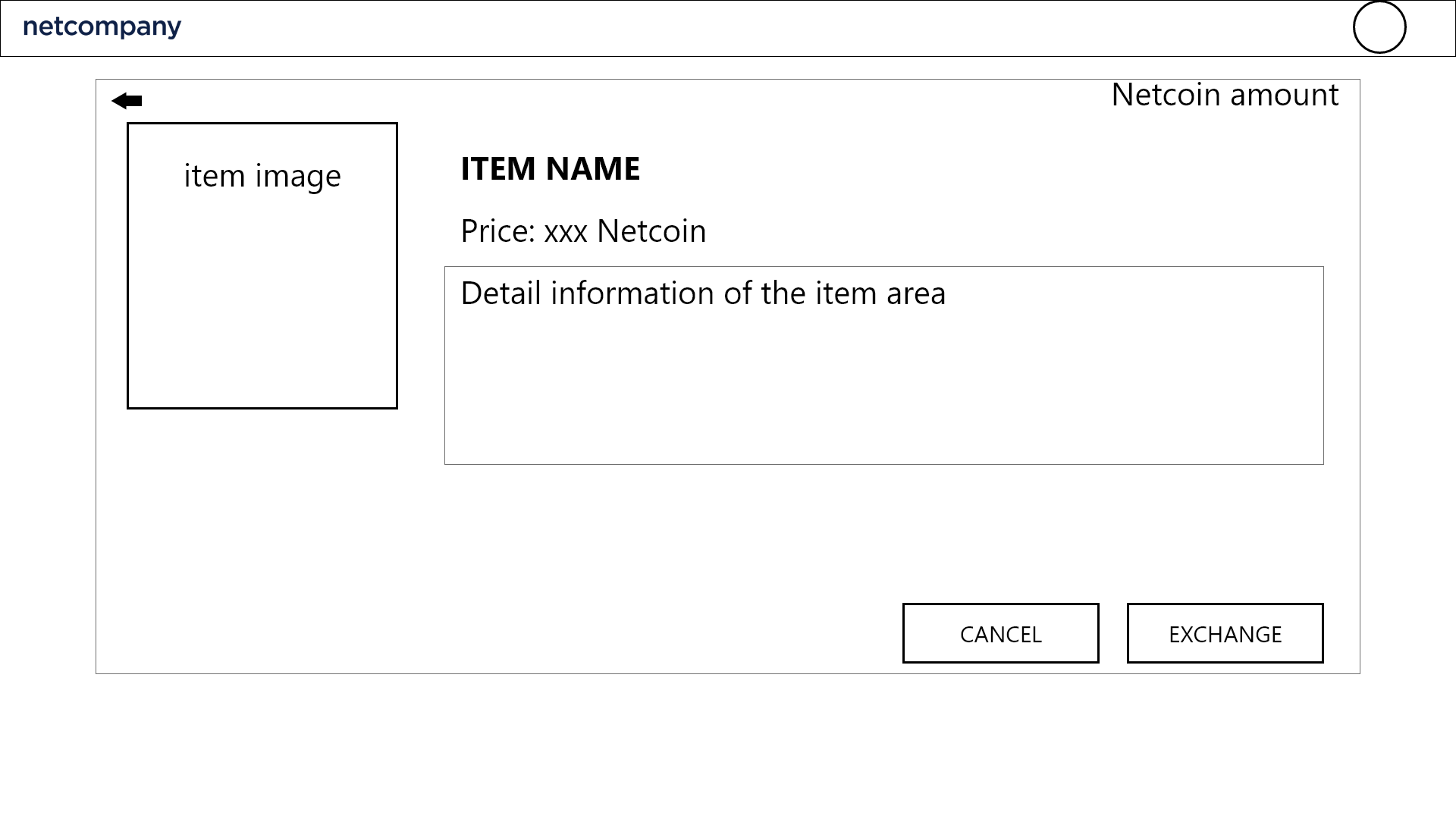
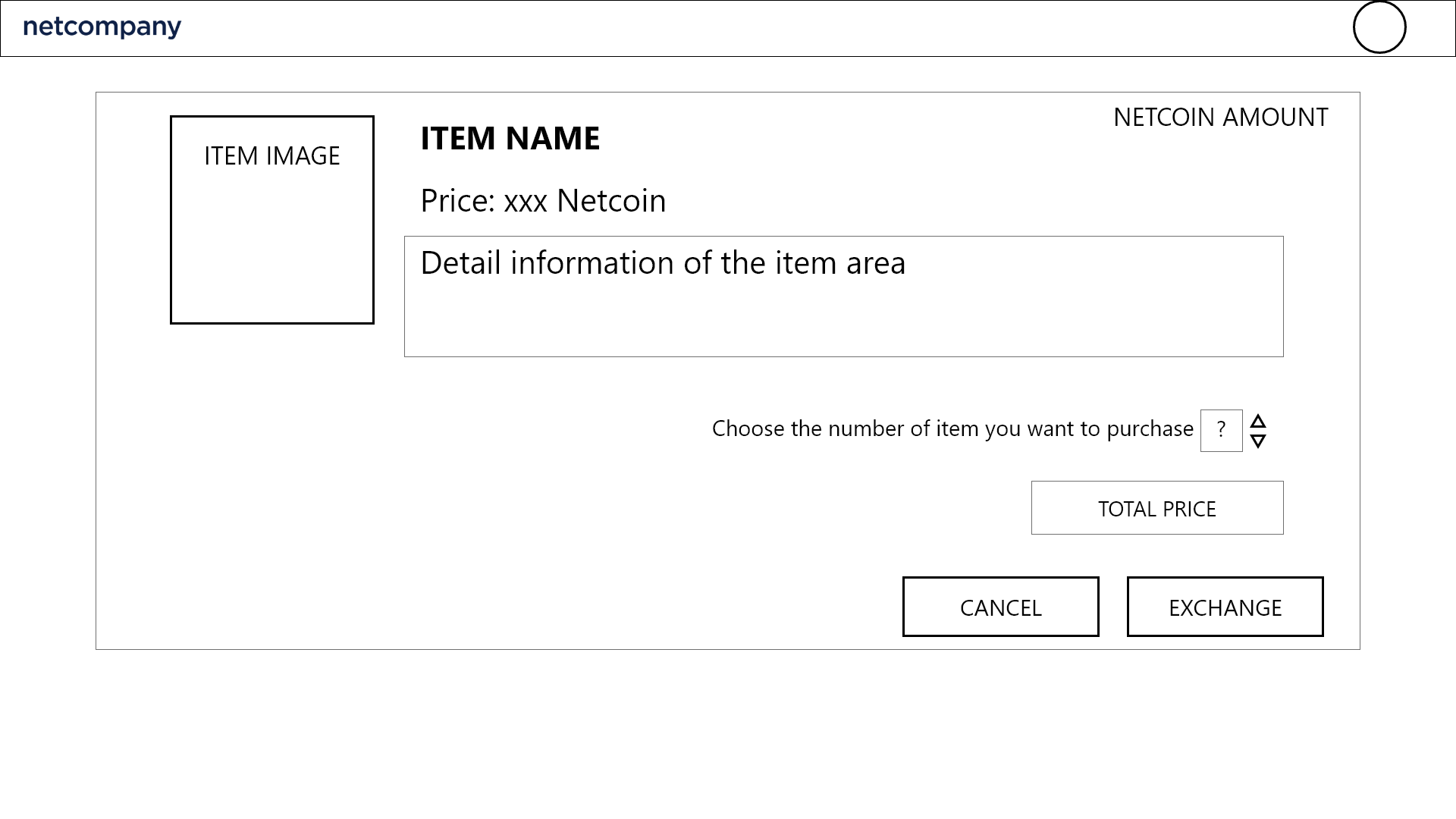


Figure 32 - Exchanging system - Item information

On the top right corner of this figure is the amount of Netcoin the user is having . Next is the image of the current item the user is selecting. Next is the Item name, price and the detail information of that item

Down below are the 2 buttons, on the left side is the cancel button, when clicked it will return the user to the previous page, on the right the right side is the exchange button when clicked it redirects to figure 33.

Figure 33 - Exchanging system - Item exchanging

In this figure main container, the item image and item information is the same as figure 27. Below is the item quantity the user wants to purchase.Down below is the 2 button, on the left side is the cancel button, when clicked it will return the user to the previous page, on the right the right side is the exchange button when clicked it will purchase the item if the user have enough netcoin and show error if not.

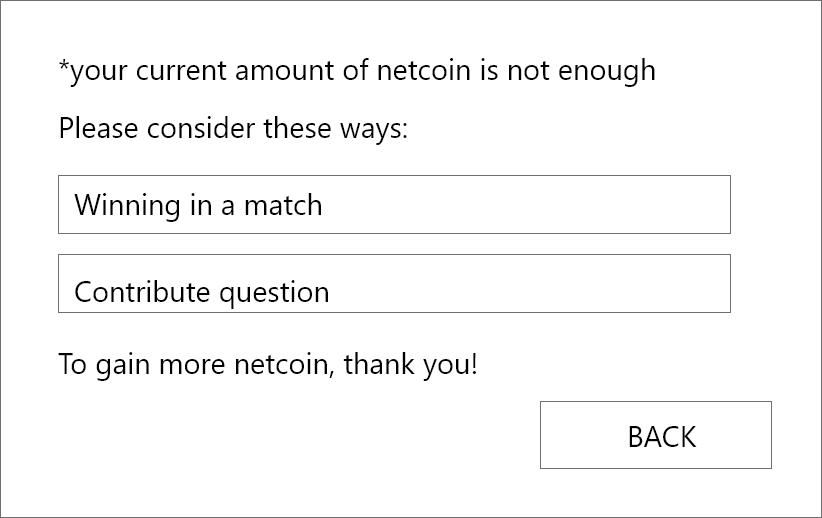


Figure 34 - Exchanging system - Netcoin is not enough

This figure appears when the user purchases an item but the netcoin is not enough, it shows the warning about netcoin is not enough and suggests ways the user can earn more netcoin. There is a back button, when clicked it will return the user to the previous page.

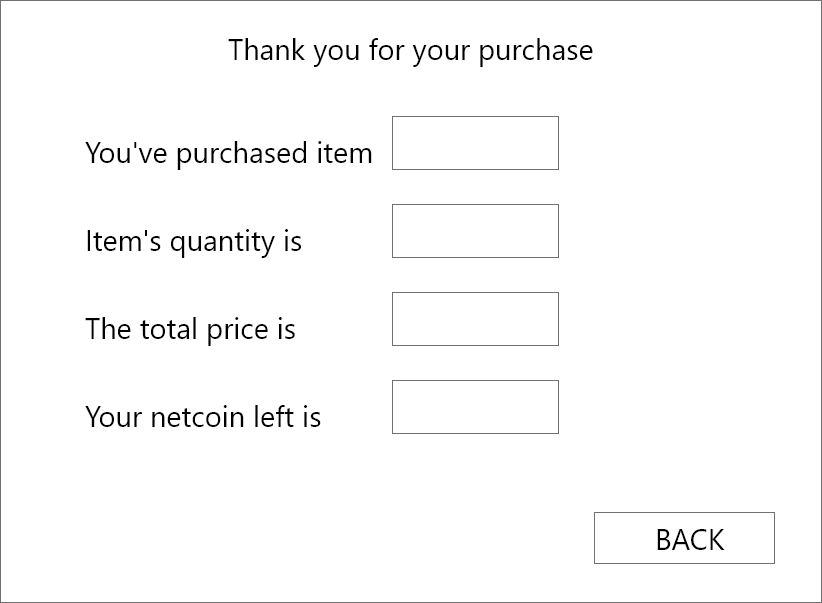


Figure 35 - Exchanging system - Netcoin is enough

This figure appears when the user purchases an item and the netcoin enough, it shows the information of the exchange. There is a back button, when clicked it will return the user to the previous page.

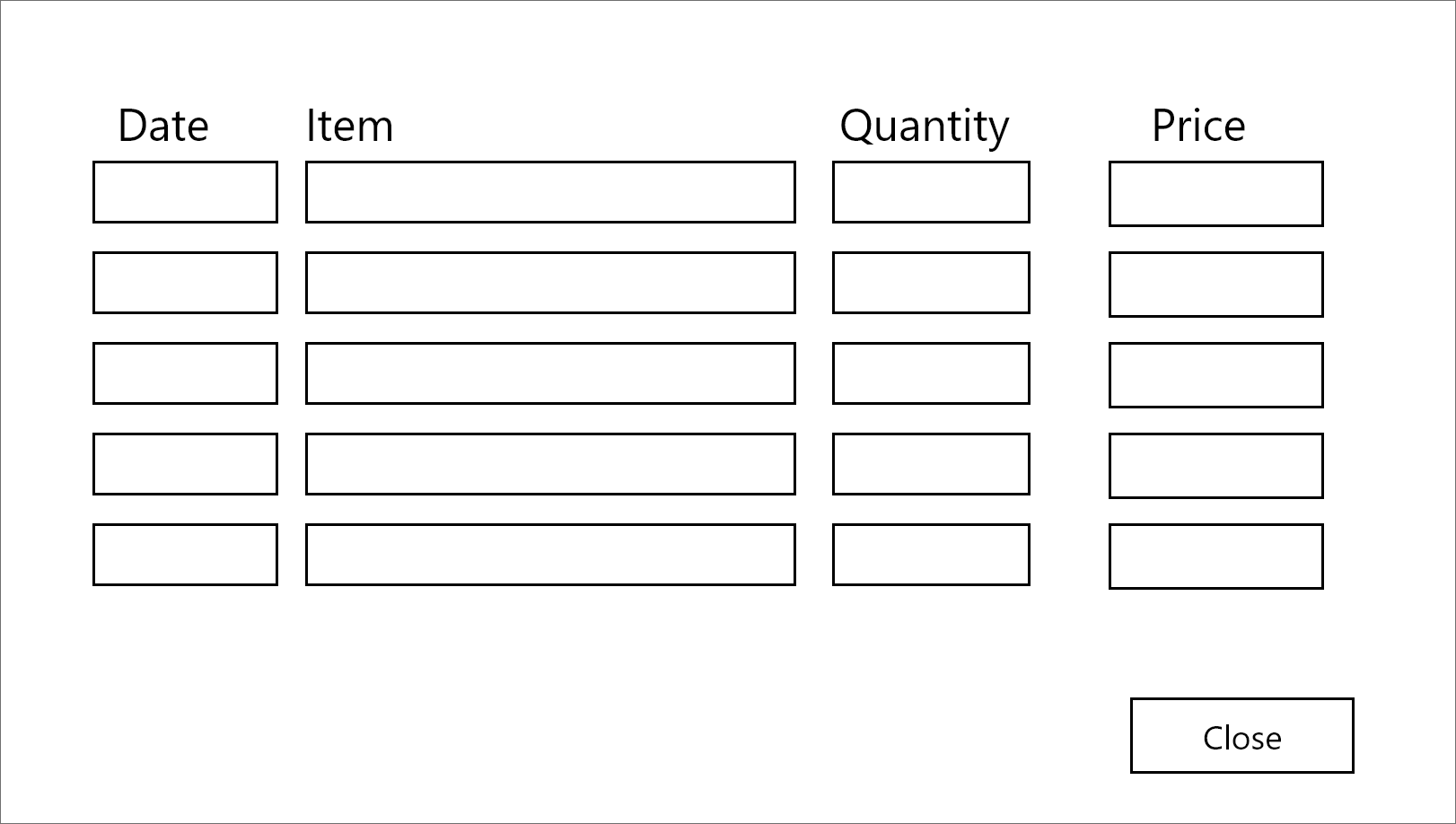


Figure 36 - Exchanging system - purchase history

This figure appears when the user clicks on the purchase history button, the main containers of this figure consists of the information of the user’s purchase history, including the item name, the date it was purchased, the quantity of that item the user purchased and the total price. There is a close button, when clicked it closes this pop-up and returns to the previous page.

## 

## Profile management (Player)

In the figure 28, when click on the information change button, it will redirect to this information layout figure

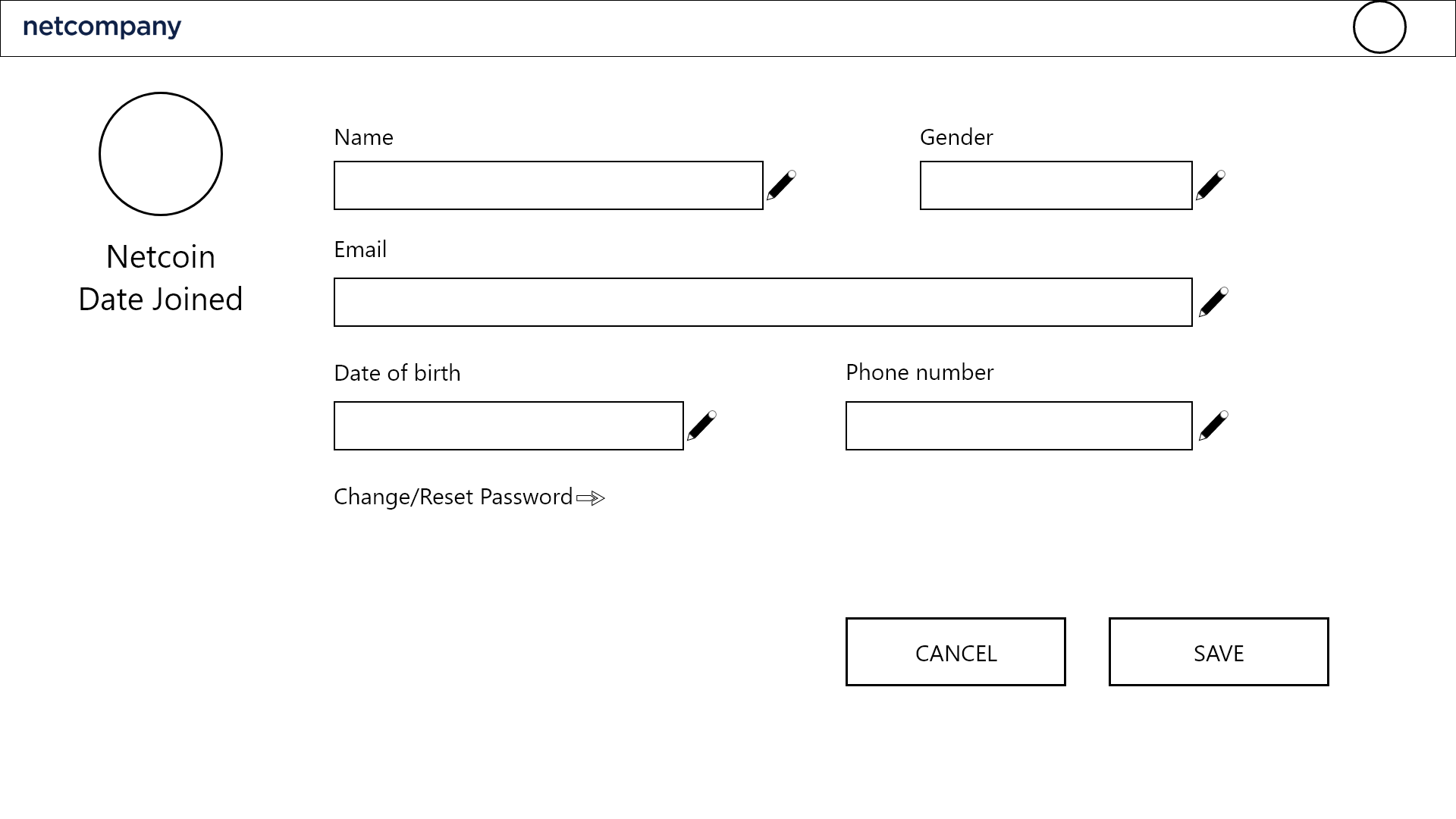


FIgure 37 - Profile management - Profile editing

In the main container of this figure, First is the avatar and the name of the user. Next is editable detailed and personal information of the user including change/reset password when clicked it redirects to change/reset password section.Down below is the 2 button, on the left side is the cancel button, when clicked it will return the user to the previous page, on the right the right side is the exchange button when clicked it save all information.

## 

## Gameplay (Player)

In figure 24, when the user clicks on the join the waiting room button it will redirect to this figure

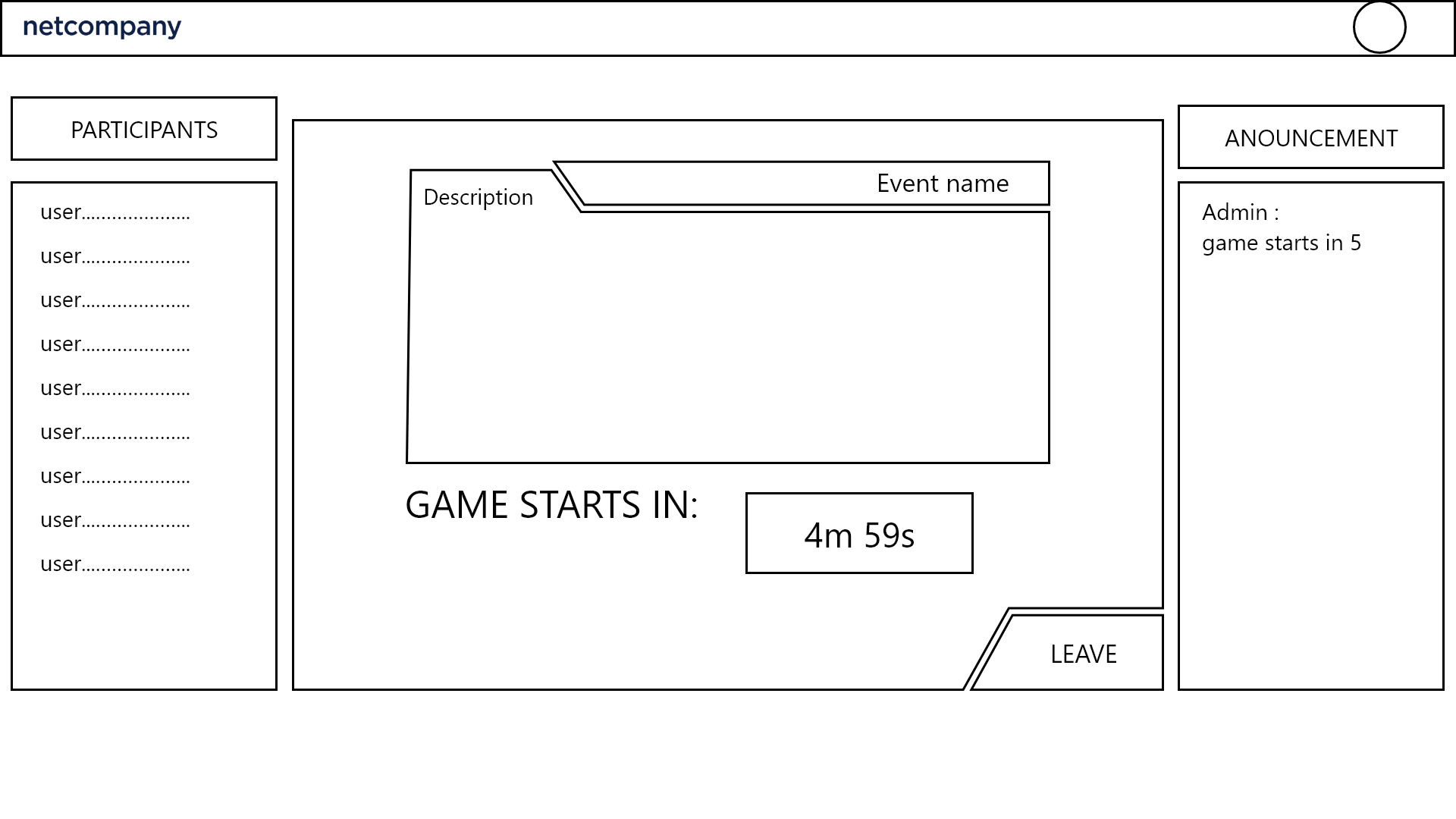


Figure 38 - Gameplay - Event waiting room

In the main container of this figure, on the left side is the participants list, it shows the name of users who already join the waiting room. On the right side is the announcement box,it shows the announcement from the host.

In the middle is the information of the event, including the name of that event, short description and a countdown timer until the game starts. When the timer hits 0, it will change to a join button. When clicked, the join button will join the user to their playing screen (see figure 39).

Last is the leave button, when clicked it will return the user to the previous page.

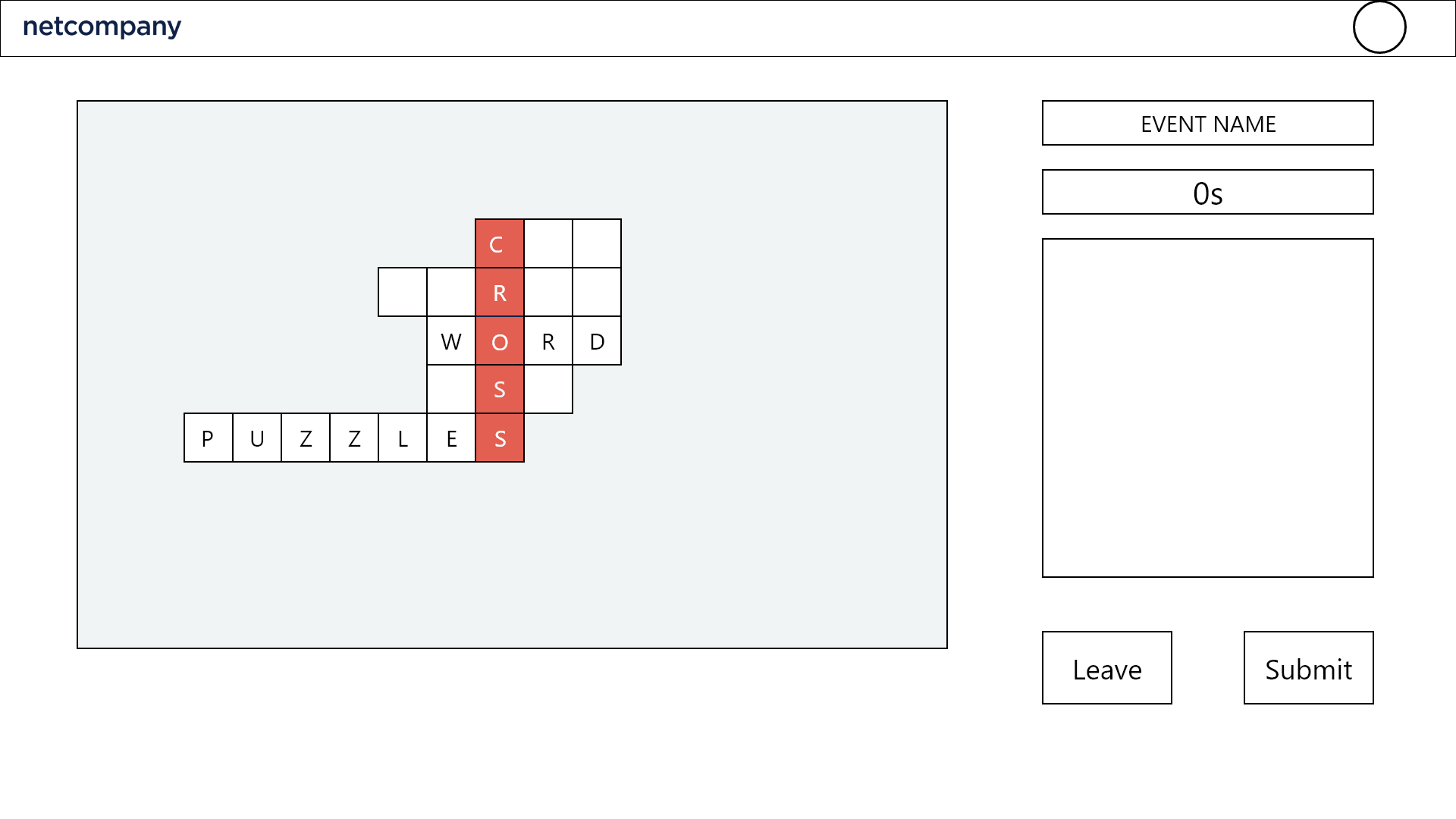


Figure 39 - Gameplay - Playing room

In the main container of this figure, on the left side is the crossword puzzle of the game the user is playing, on the right side is the information of the event and the puzzle.

The information of the event including the name of that event and a countdown timer to when it ends, when it hits 0, the screen is clocked, the answer will be submitted automatically and a pop-up will show up to announce.

The information of a particular question is shown in the question box when the user clicks on any horizontal line in the puzzle.

Last are 2 buttons, the first is the leave button, when clicked it will pop a arlert if the player wants to leave, last is the submit button, when clicked it will send the puzzle and the user’s answer to the admin and at this time the answer is not editable anymore.

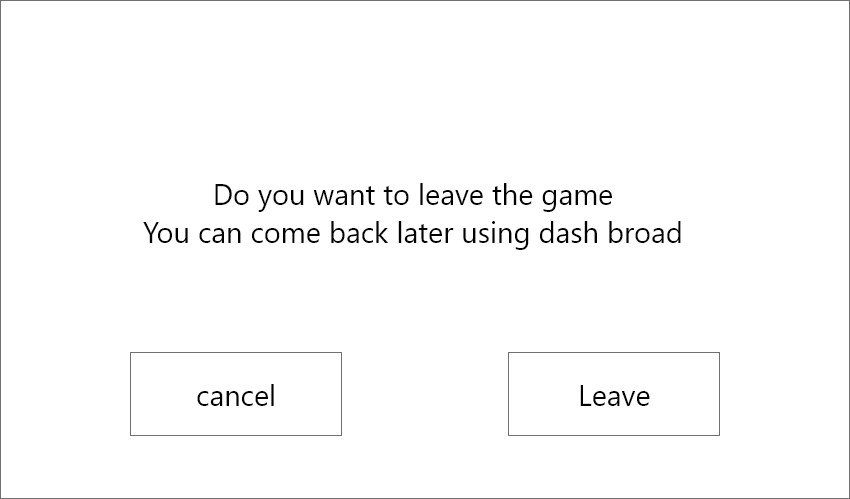


figure 40-Gameplay - Leave confirm pop-up

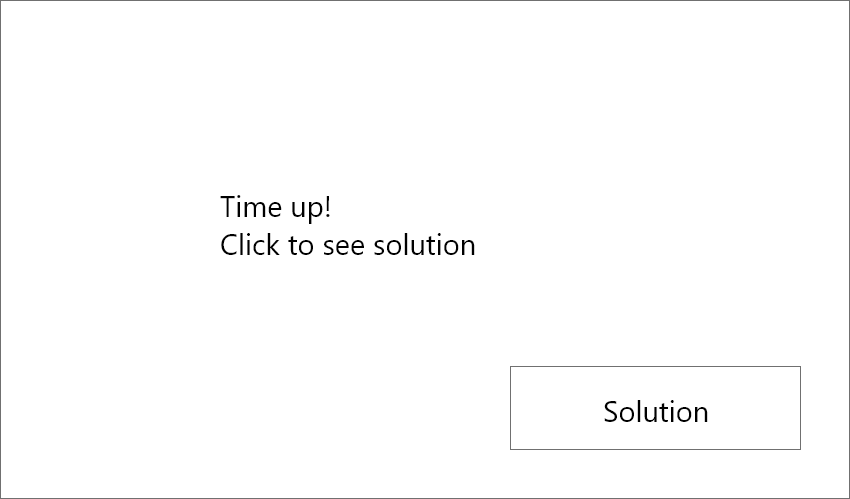
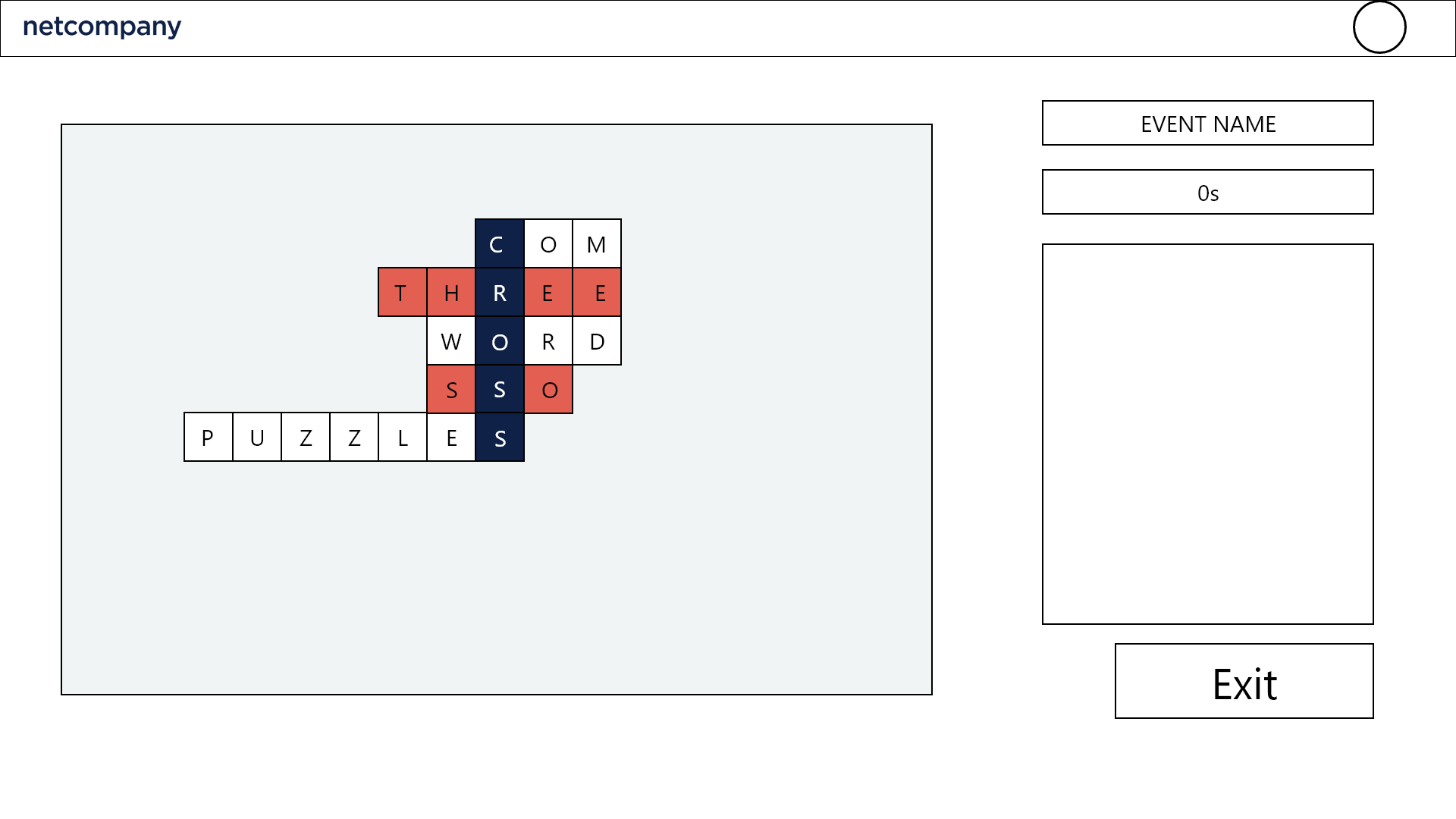


Figure 41 - Gameplay - Time up pop-up

When the countdown timer in figure 38 hits 0, the screen is clocked, the answer will be submitted automatically and this figure will show up to announce. There is a solution button, when clicked, it redirects to the solution figure (see figure 40).

Figure 42 - Gameplay - Solution

The main container in this figure is the same as figure 38, but the puzzle now will also show the solution with it. It also highlights the player’s unanswered lines. There is an exit button, when clicked it will redirect to the match result figure (see figure 41).

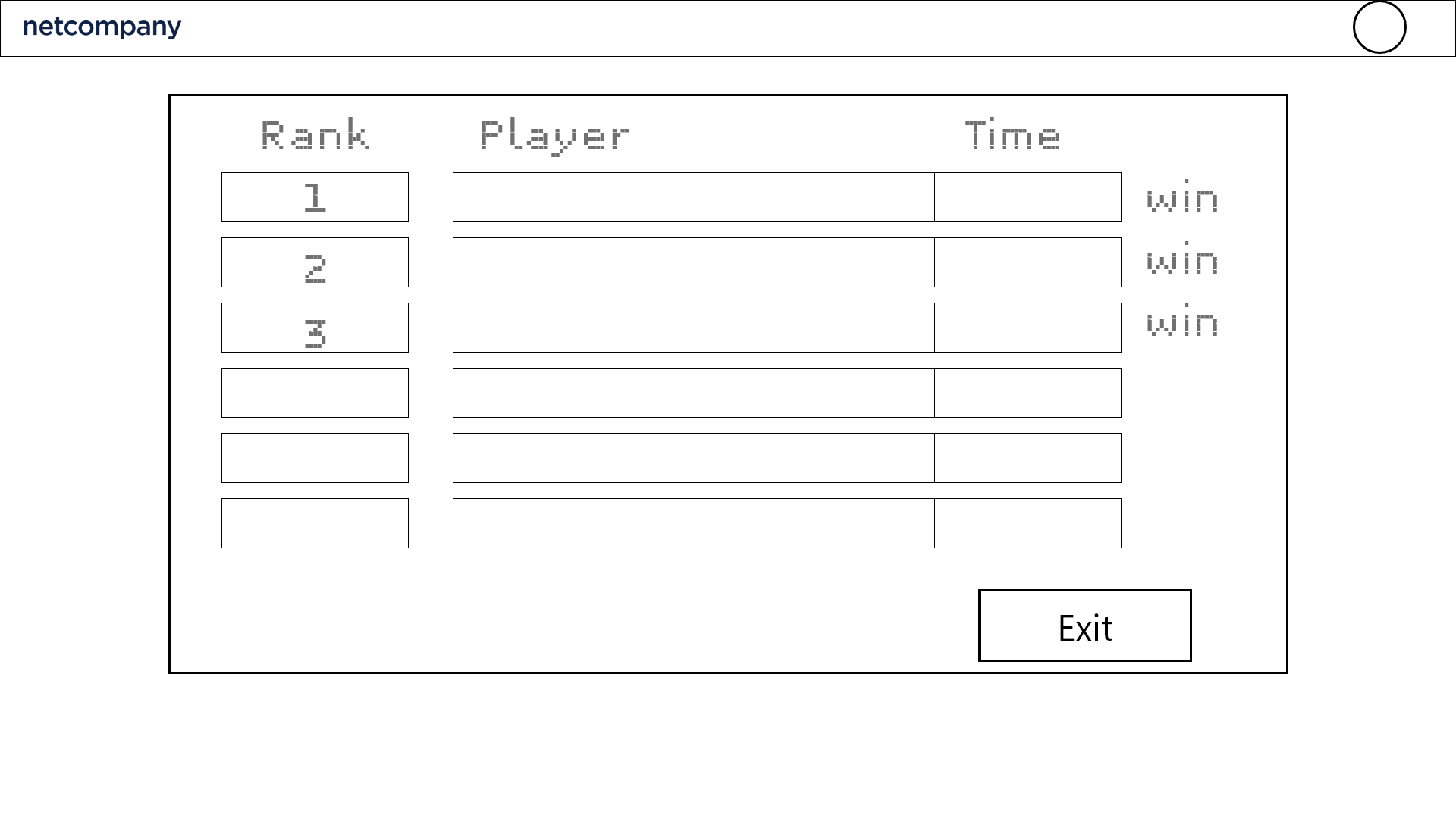


FIgure 43 - Gameplay - Match result

When the game ends, the ranking list of the participants in that game is shown in the form of a list view. The rank of a user is calculated based on the number of questions, the time spent and the difficulty of those questions that the user answered.

There is an Exit button, when clicked it returns the user to the dashboard (see figure 24).

# Standard Presentation Layer

(nội dung chính của phần 3 là gì?)

This section describes all of the global components available in the web application, including both admin and user (player) interface. These components are considered global due to their generic functionality and reusable design. Some of these components will be custom implementations.

## Graph Components

### Functional Description

A number of custom-built graph components will be implemented for the admin dashboard screen.

The following graph components will be available in the dashboard.

#### Line Chart

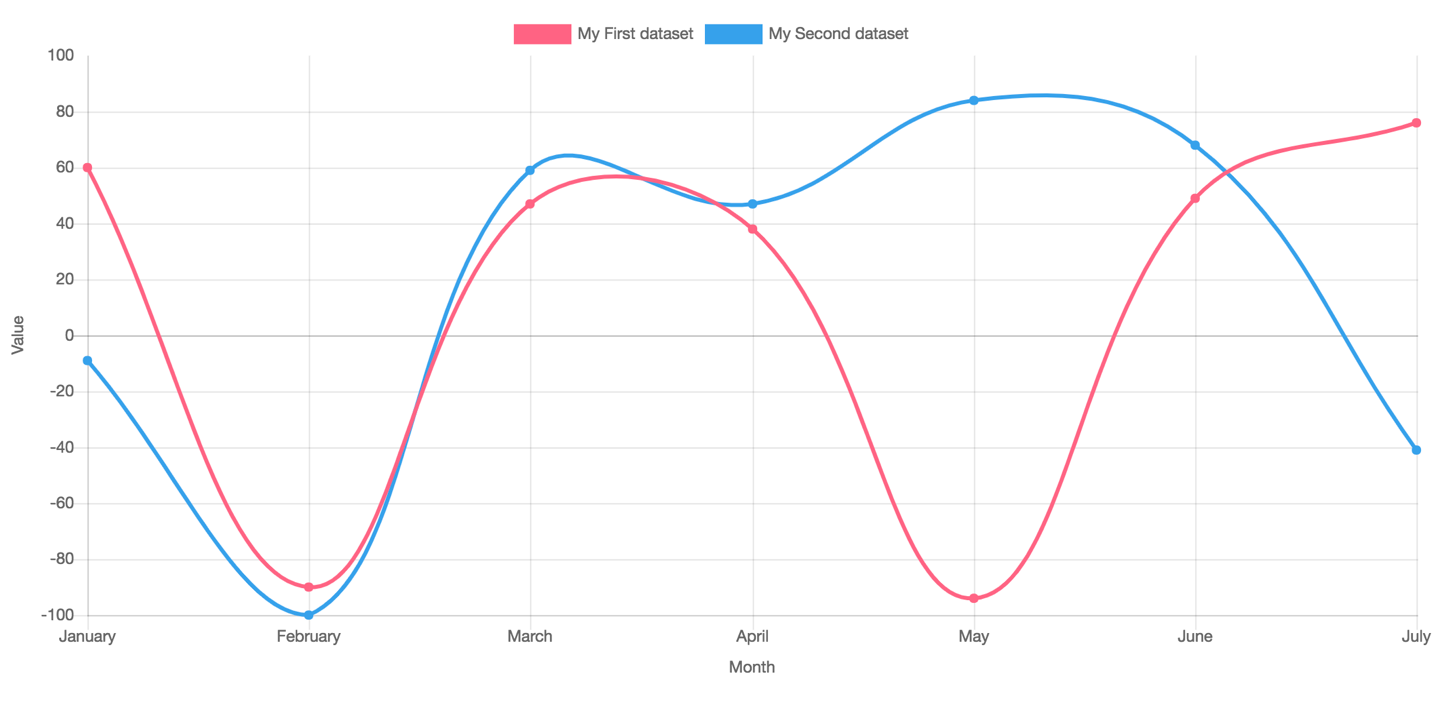


FIgure 42 - Line Chart component

A basic line chart will be implemented. Once a line chart component has been added to a placeholder on a page, the editor will then have to select a data source in the admin dashboard statistics menu and fill out the required properties for a Chart Data Source Type.

The visitor is able to narrow the timespan along the X-axis using two datetime pickers in the top of the graph.

#### Bar Chart

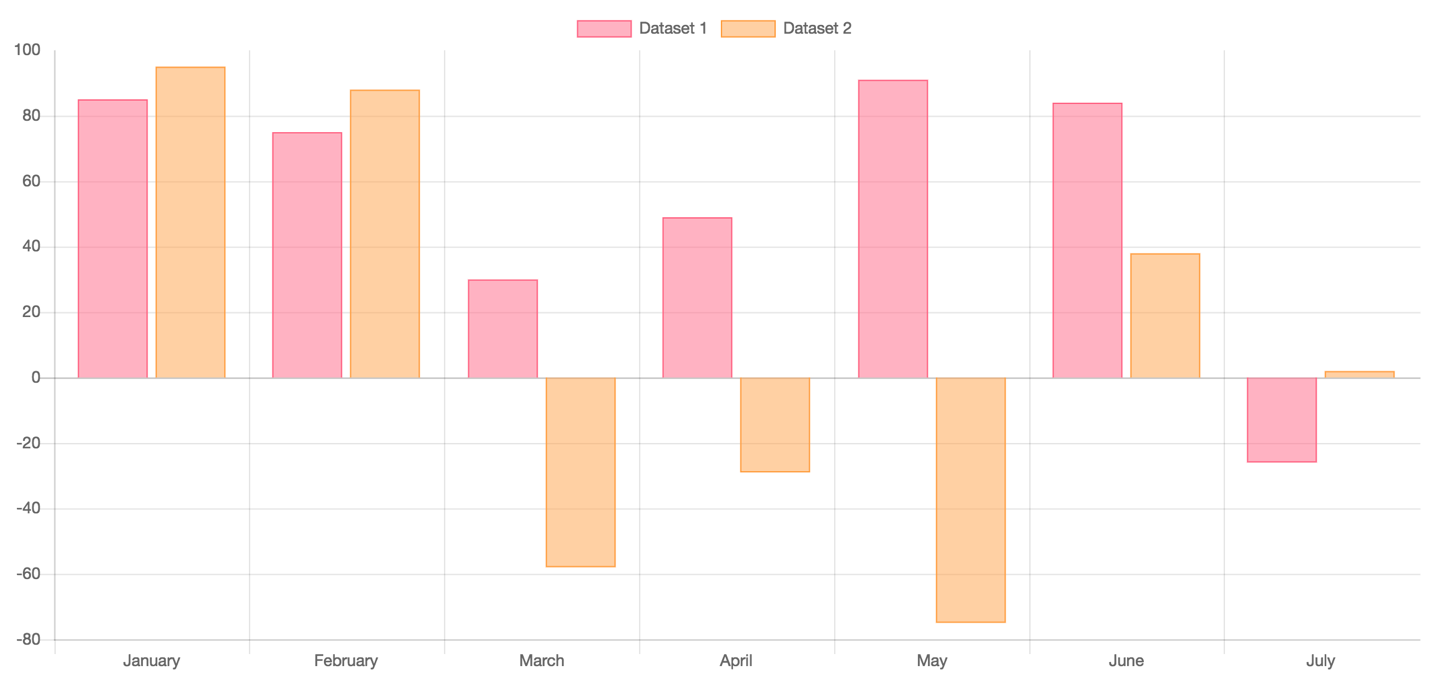


Figure 43 - Bar Chart component

A basic bar chart will be implemented. Once a bar chart component has been added to a placeholder on a page, the editor will then have to select a data source in the admin dashboard statistics menu, and fill out the required properties for a Chart Data Source Type. A bar chart can have both negative and positive value points on the Y-axis scale as shown in Figure 43.

When more than one datasets is plotted, the visitor will be able to toggle the label in the top of the chart to show/hide the individual bars in the chart.

#### Stacked Bar Chart

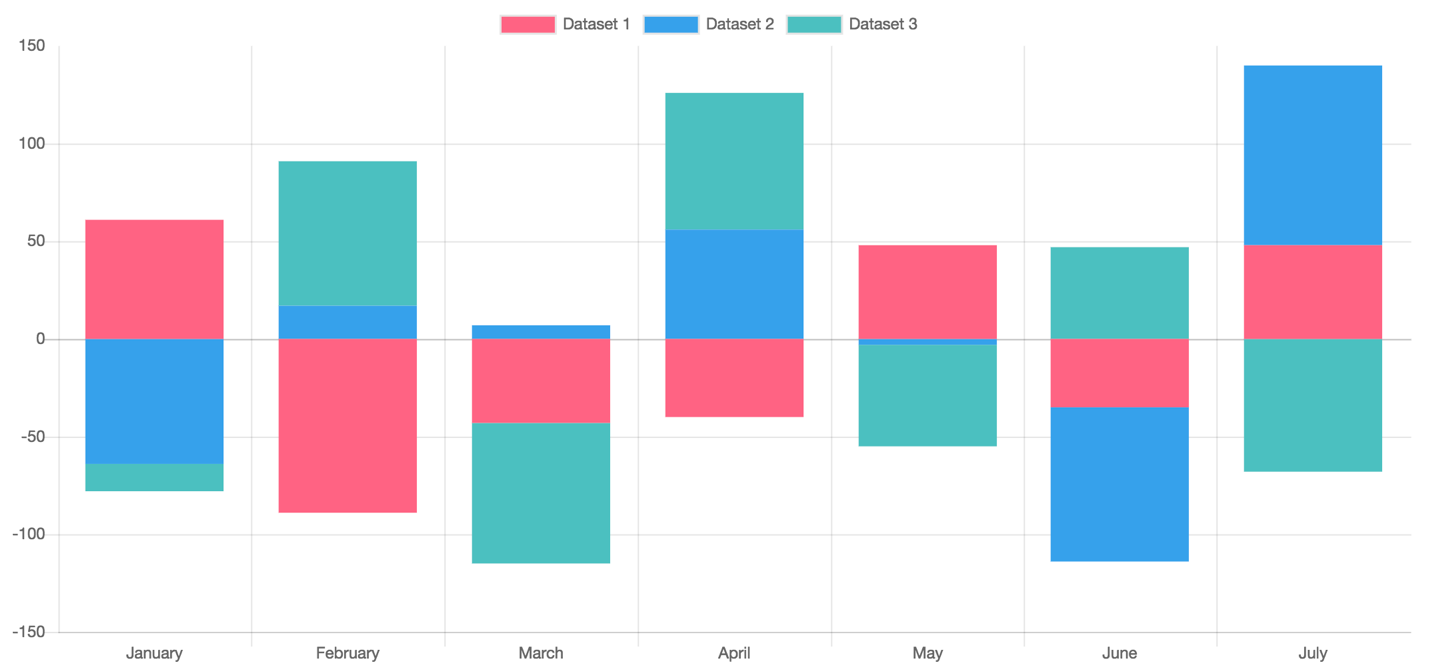


Figure 44 - Stacked Bar Chart component

Similar to the basic bar chart, the stacked bar chart is able to chart both negative and positive value points on the Y-axis scales. However, when charting multiple datasets, each set will be stacked on each other as seen in Figure 44.

The visitor is able to narrow the timespan along the X-axis using two datetime pickers in the top of the graph.

#### Pie Chart

FIgure 45 - Pie Chart component

A pie chart will be implemented. Once a pie chart component has been added to a placeholder on a page, the editor will then have to select a data source in the admin dashboard statistics menu, and fill out the required properties for a Chart Data Source Type. Each slice in the pie can be toggled to hide/show by clicking the labels at the top of the graph.

The visitor is able to narrow the timespan along the X-axis using two datetime pickers in the top of the graph.

#### Embedded Google Map

Embedded Google maps with no styling for showing location of the upcoming event or the location of Netcompany. A screenshot of an unstyled Google Maps component is shown in Figure 46.

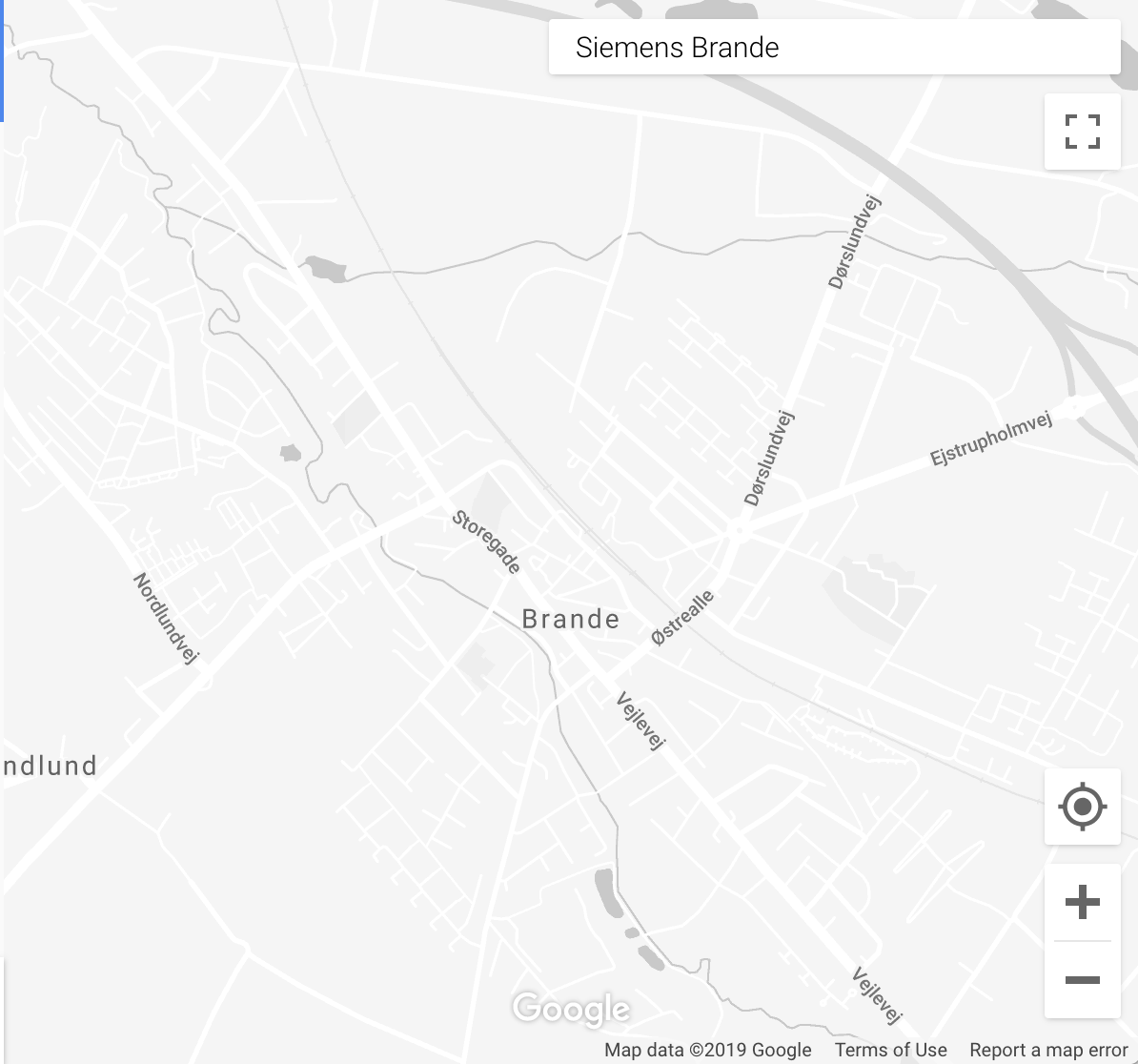


Figure 46 - Google Maps component

### Integration Design

The data feed used for the custom graph components will be a single data warehouse. Data will be read on demand based on views provided. Details on the integration perspective can be found in the Integration Strategy section.

#### Data Validation Rules

When data is presented to the customer, the quality of the data is validated based on if the data is present and if the connection to MongoDB is working. If this is not the case, a banner alert is displayed on the page. See the Banner/Alert Component section for more details on the banners.

## Banner/Alert Component

### Functional Description

The website will have the option of showing persistent (i.e. still visible when scrolling) notification banners below the navigation pane or inside some particular placeholder, which can be used to inform the users about any larger issues that might be affecting many users. External users can then choose to dismiss the banners as they see fit, assuming that the option for dismissing a particular banner is not disabled.

Internal administrators will be able to create and administer such notification banners / alerts through the dashboard. A banner will contain a simple text notification and it will be possible to choose the severity of the banner based on some templates, resulting in different visuals and icons, such as a yellow banner for a notification of medium importance or a red banner for a critical notification which is not dismissible. A banner can be made visible side-wide or on a specific page, or it can be further customized to be customer specific or site specific through personalization based on available customer master data.

## Dropdown Selection Component

### Functional Description

The Dropdown component as seen in Figure 47 enables the customer to choose the context of information being displayed on a page (when available). Click on the dropdown and that will drop a list of context for the user to choose.



Figure 47 -Selector component

## Table Component

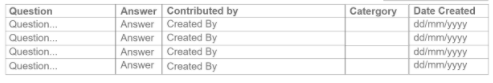


Figure 48 - Table component

The table component is a standard HTML table, however on mobile devices it can scroll horizontally when viewport is too small.

# Non-Functional Requirements

(Ac có yêu cầu gì ko? Hay nhóm tự quyết định)

## Performance

### Capacity

Based on the requirements of the game to make sure the services have enough spare capacity to handle any likely increases in workload, and enough buffer capacity to absorb normal workload spikes, between planning iterations and to be available for a quite amount of users to be access at the same time, but still be able to optimized the given resources and prevent wasting on those resources. Also estimated and calculated based on various types of previous events and average to maximum number of participants, the number from 200-500 end-users is decided to be the capacity of NCP.

## Usability

Since the project scope is small, usage is only for students and HR staff so the design of the interface will ensure the learnability, efficiency and satisfaction level to increase overtime.

Throughout the website, there will be hints banner and notifications banner (see section 3.2) for further user instruction, to help reach their goals more effectively and quickly.

The following activities will have instruction banners:

User (Player):

* Question contribution
* Gameplay
* Reward exchange

HR staff:

* Create a statistical component
* Approve / decline a question
* Edit / Create a particular item, for example a game event

## Security

HTTPS protects the privacy and security of your users

HTTPS prevents intruders from being able to passively listen to communications between your websites and your users. One common misconception about HTTPS is that the only websites that need HTTPS are those that handle sensitive communications

Therefore, the application will be using HTTPS instead of HTTP for data encryption and privacy. Users could be assured about their personal information and communication across the Internet.

The system will set roles for the users so just only the admin role could see the user public information. Users can not see other user data. The NCP could secure users from unauthorized access. Only the HR staff have access to the admin dashboard for management of their choice and the overview of the NCP private data.

## Visual Expression

### Mobile Friendly

#### Responsiveness

The Responsiveness will be customized according to the size of the screen, those will be responsive and adapt to three different form factors: Mobile, Tablet and Desktop. The responsive will be based on the D0100 document.

#### Performance on Mobile Devices

In order to optimize performance on mobile devices it might be necessary to hide certain components, e.g. complex graphs that rely on very large datasets for rendering the chart or optimize images to get the most out of the site storage and avoid the bandwidth limit. Otherwise it can customize it individually to hide some unnecessary components when responsive in form Mobile and Tablet.

#### Supported Browsers

The application will support the following browsers and browser versions, as defined during the clarification workshops.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Mozilla Firefox | Google Chrome | Apple Safari | Apple Safari for IOS | Google Chrome for Android |
| 83 | 87 | 14.0.1 | 14.0 | 87 |

In general, the product will focus on the latest browser version of the above mentioned browser vendors, which will cover the requirements.

### Design Guideline

The detailed design of Trading Vision Project (TVP) will be further explained and based on the D0100 document.

# High-level Software Architecture

(vậy nhóm đang làm theo mô hình nào?)

## Architectural Goals and Constraints

This is the breakdown of the code’s layers, how the NCP principle has been implemented and how the architecture meets the demands of the project.

### Architectural Model

The chosen implementation for the solution is based on the MVC (Model - View - Controller) architectural pattern.

The model divides the solution into three components: **Model**, **View** and **Controller**, for which all dependencies flow between components interacting with the clients.

Each component has a well-defined role as described in section **5.1.2** - **5.1.4**.

Each component will be further divided into logically separated classes and smaller components.

A sample illustration of this modular approach is found below.

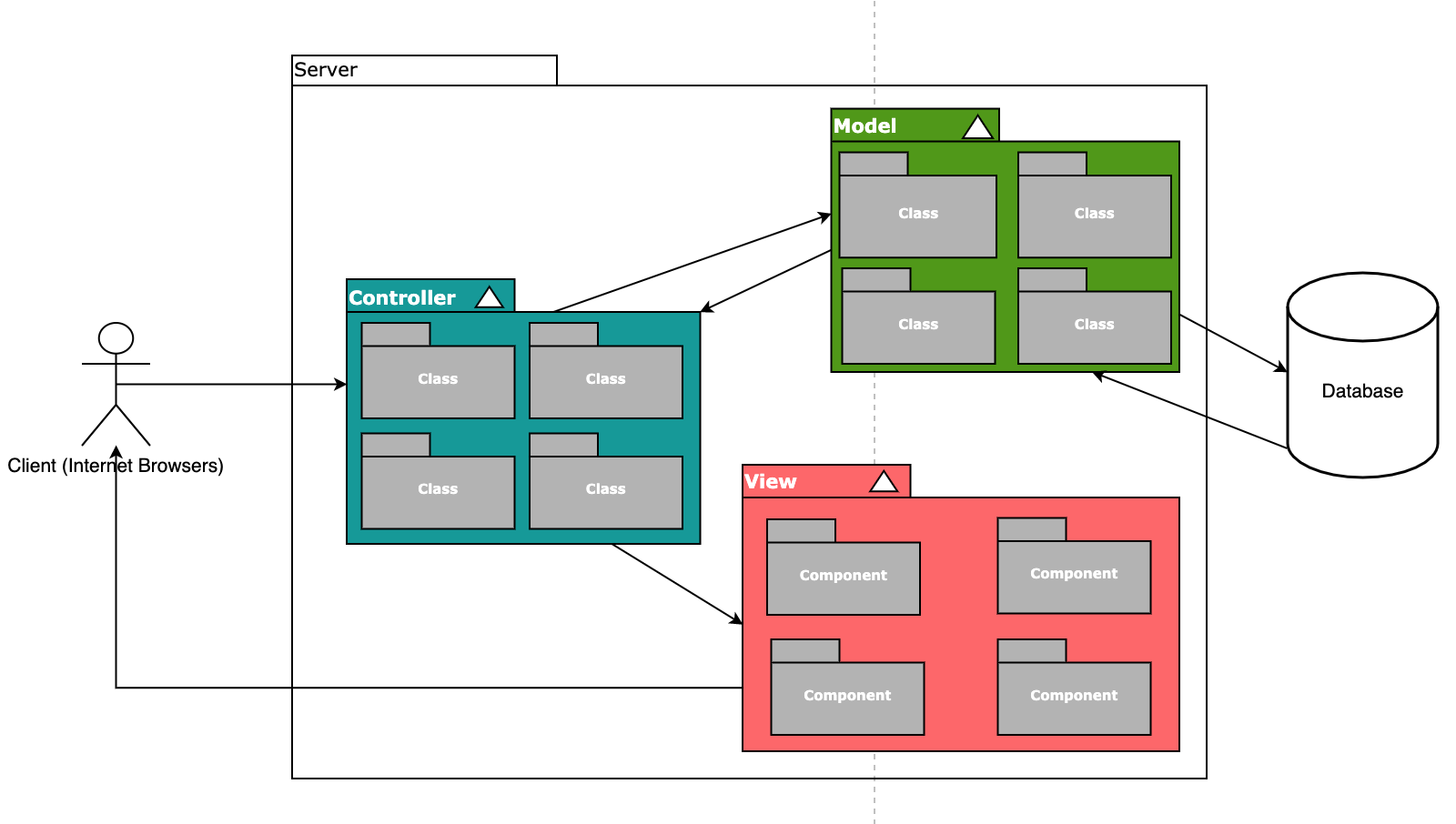


FIgure 49 - Architectural Model

### Model

The model component stores data and its related logic. It represents data that is being transferred between controller components or any other related business logic (application business logic, validation logic, and database access logic)

It responds to the request from the views and also responds to instructions from the controller to update itself.

The model consists of many classes, each model is the representative of an entity / a collection within the database

### View

The view component represents the actual websites for the solution and consists of many smaller components.

Views request the model to give information so that it resents the output presentation to the user and are created by the data collected from the model data.

A web page / screen is generated by many components of the views.

### Controller

The controller component is responsible for controlling the way that a user interacts with an MVC application, it determines what response to send back to a user when a user makes a browser request.

It also sends commands to the model to update its state and sends commands to its associated view to change the view's presentation.

The controller consists of many classes. Each model will have its own corresponding controller.

## Development Principles

The detailed design will be broken into user stories that can be delivered by the developers throughout the sprint in the project. The developer has to follow the MVC architecture as described in section **5.1**. Furthermore, there will be two full detailed documents that serve as background knowledge for any development: “**O0500 - Software Architecture**” and “**DD160 - Programming Guidelines**”. The two documents help the developer to know where to place new integrations, classes and code in the code repository to follow a unified code quality.

The MVC architecture framework helps to separate the application into three main logical components, which are built to handle specific development aspects of the application. There is a strict rule between the three components means that a component can not be removed or replaced, without affecting the solution at all. This creates a very structured code base that is easy to maintain and further develop over time.

Pull requests and code reviews are used to ensure high code quality. This means that even though every single developer is responsible for writing code with quality, the combined solution is shared responsibility, and the approver of a pull request should stand behind the code that will go into test and later on production. Pull requests give all developers the chance to review and comment on code changes, before they are merged into a shared branch. The solution can then be discussed and resolved by any member of the team (see Figure 41).

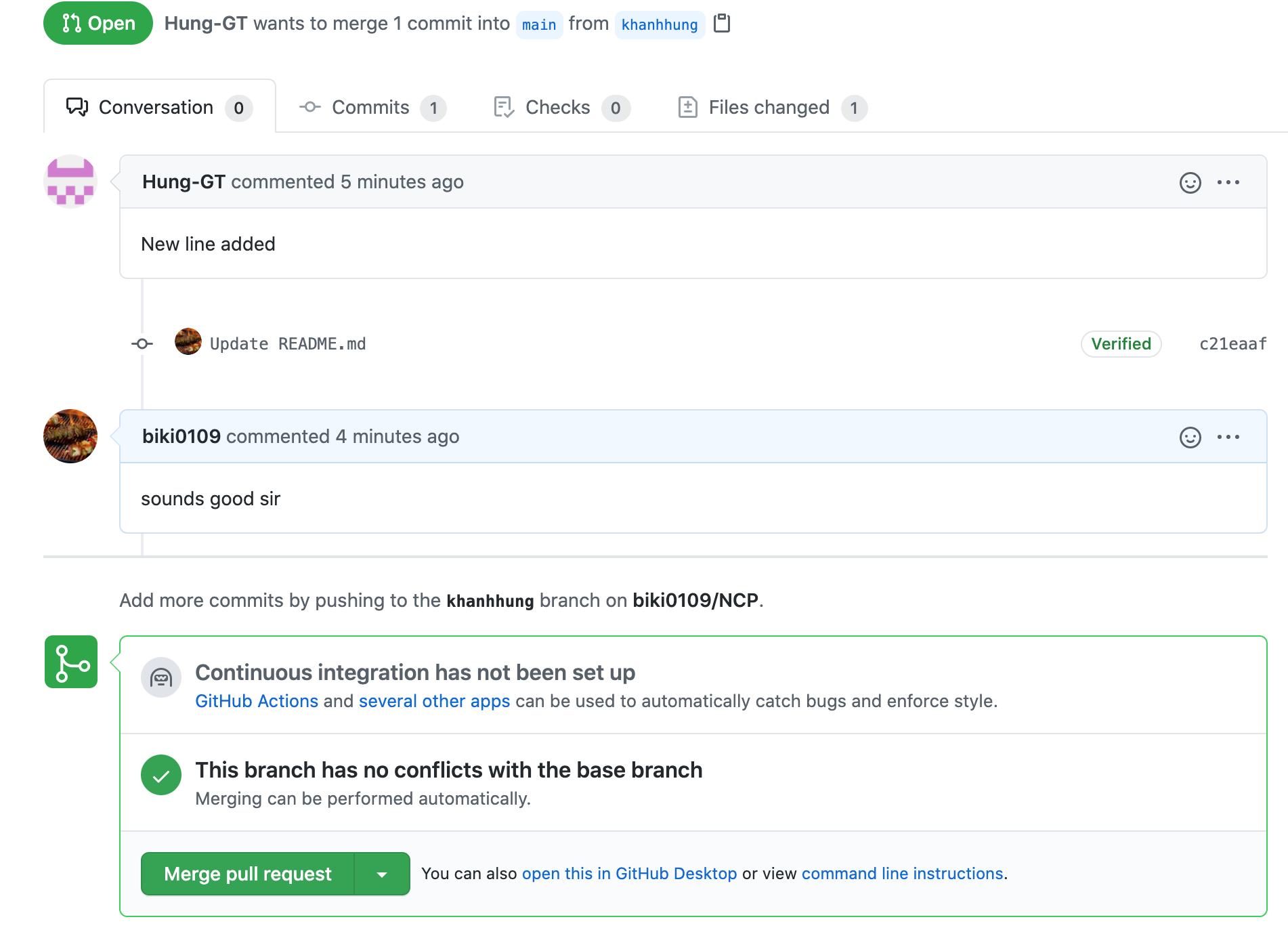


Figure 50 - Example of a comment to a pull request in Github

Criteria can be set up to ensure that pull requests are not completed, before somebody in the team has reviewed the solution and all comments are resolved. If the solution uses continuous integration, a criteria could be that the build of the new code must succeed before the code is merged in. This also ensures that no developer in the team is blocked due to a failing build in the common branch.

Naming conventions will be conducted and explained later on in the “**DD160 - Programming Guidelines**”

## Deployment Perspective

The process of deploying code to Test and Production environments can be split into two processes; a build process and a release process. The developers will be using their local environment for development and using Azure server provided by Netcompany for CI/CD, deployment and further testing.

Once the code has been reviewed and brought into a common branch on Github, it will be then imported into the Azure Repository. The Azure Pipeline will then be created for setting up the CI/CD and deployment. The team, if possible, will be using Azure Container Registry provided by Netcompany for storing container images or Docker Hub for replacement (see figure 51)

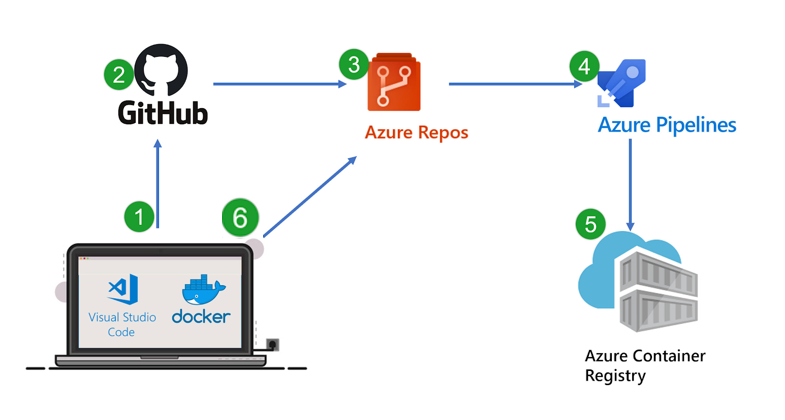


Figure 51 - Sample Flow for deployment and CI/CD

For building docker images, the NCP will be initially using three docker containers. Each for the Backend, the Frontend and the Database. Any further related docker containers will be mentioned later in this document.

### Build Process

The build process creates a **Docker Image** based on a commit / pull request in the code repository (Github) . A source branch always has a reference to a specific commit that is updated, once new changes are merged in. The setup in Azure DevOps supports multiple steps and can be further adjusted based on project needs.

Once the **Docker Image** has been set up, it is typically only modified if the source code is changed. The build process can be initiated by anyone with the appropriate access. Creating a new pipeline to run is very simple. The user can select a build package and then select a source branch to run that pipeline. Each individual step in the run process can be followed in the GUI, and the previous run can be seen in the History tab (see Figure 52).

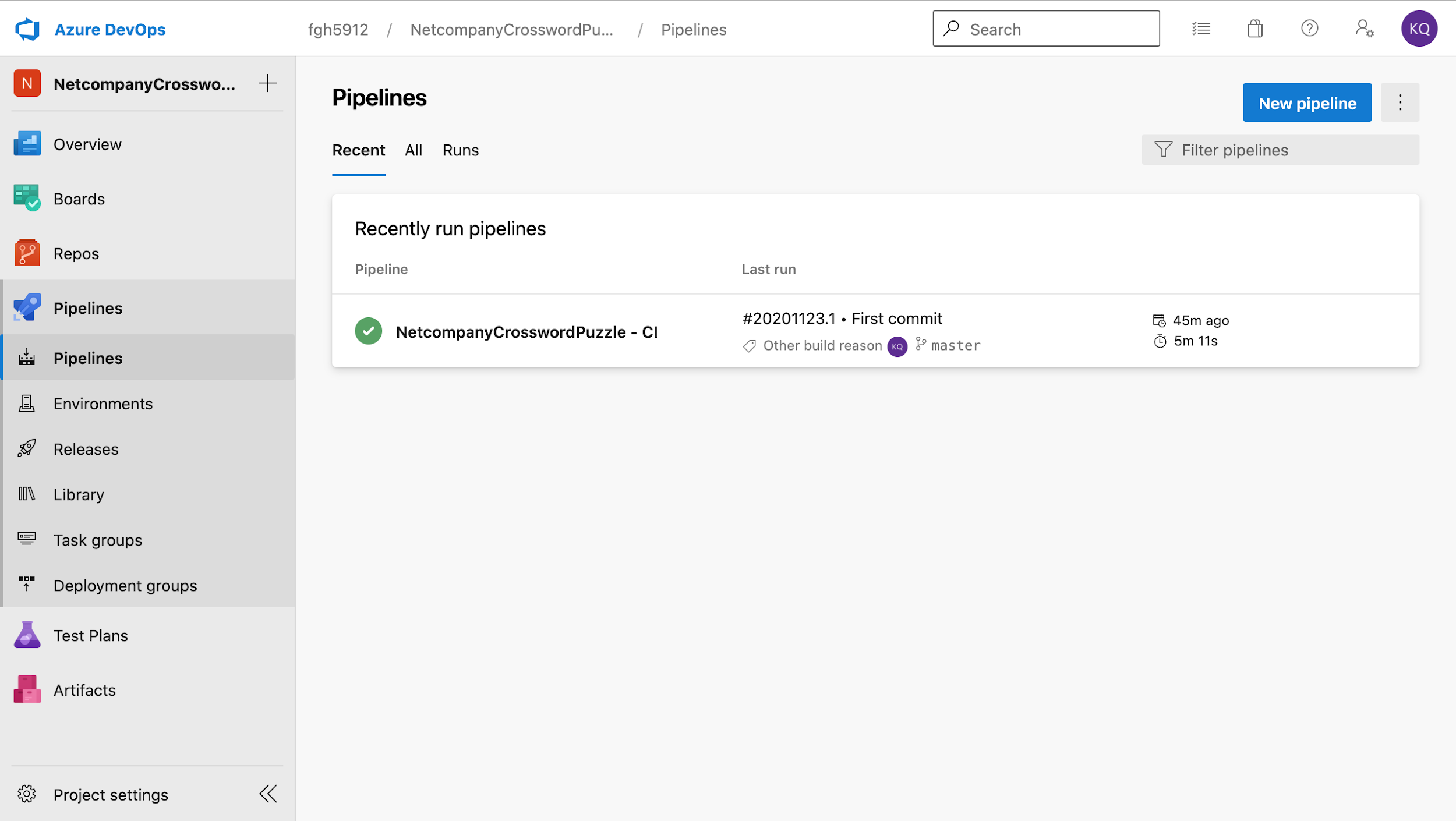


Figure 52 - Overview of run pipelines and previous runs in Azure DevOps

A sample run will have a test case summary screen showing the pass percentage (see sample figure 44).

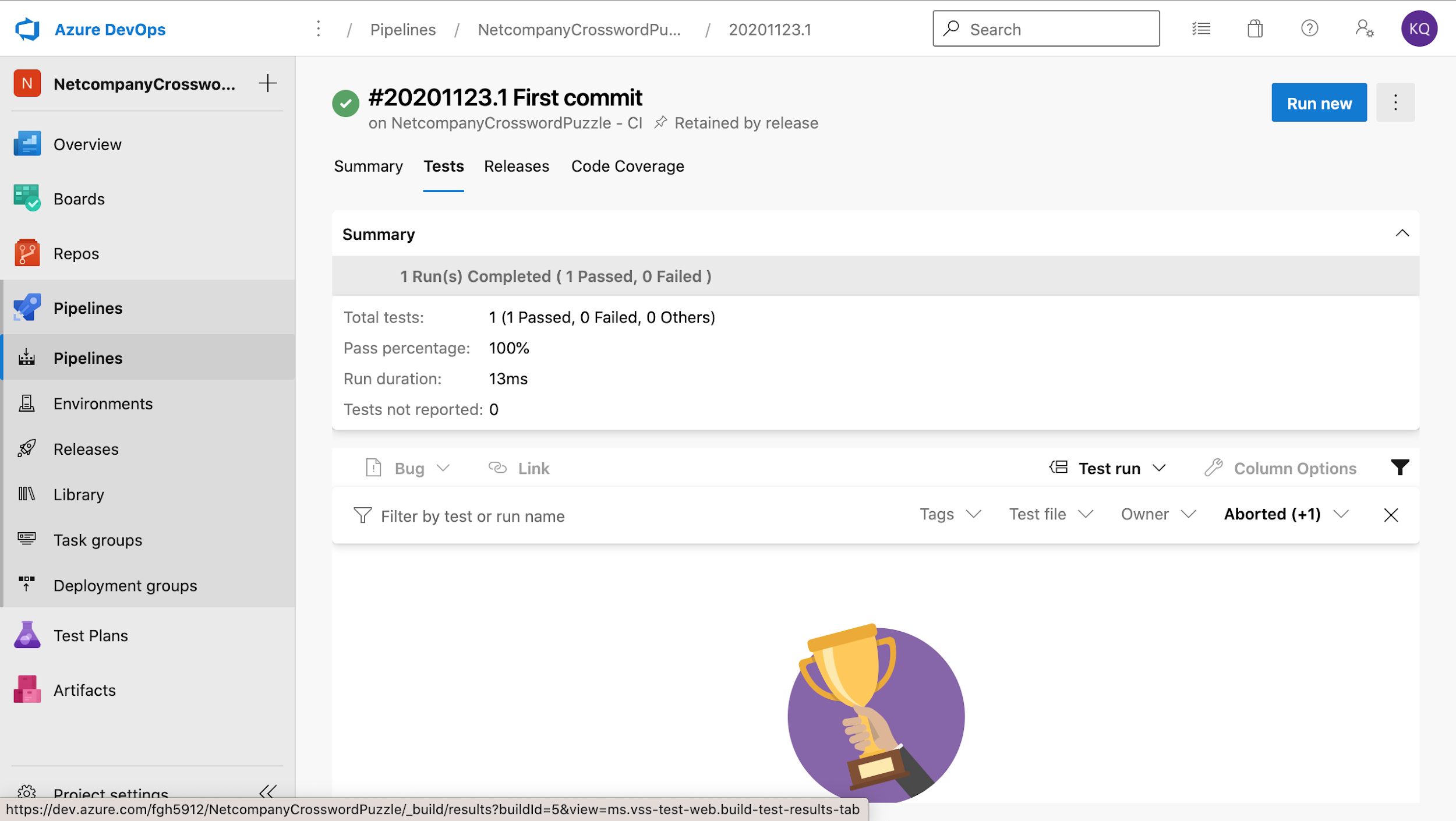


Figure 53 - Summary of test case of a run pipeline

### Release Process

The prerequisite for a release process is a successful run pipeline that can be deployed to an environment (Azure Web Service). A logical chain of environments (also called stages) can be set up in Azure DevOps, so the package is deployed to the Test environment, before it is released to the Production environment.

The release pipeline supports multiple targets. There is a standard integration for deploying the build package directly to an Azure App Service. The template can be seen in Figure 54.

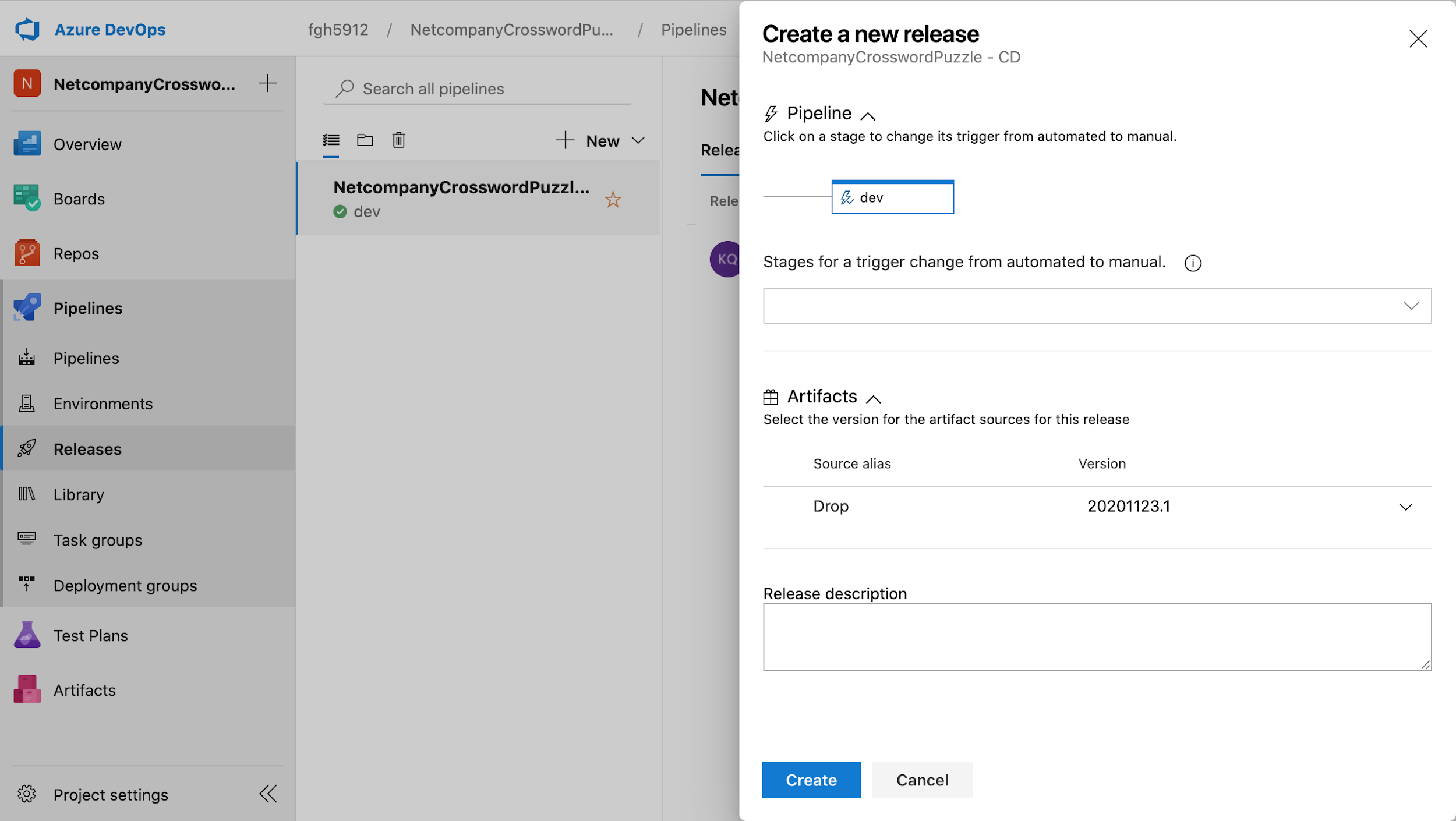


Figure 54 - Setting up the release pipeline from Azure DevOps to Azure App Service

The stages can be configured with separate and/or shared variables that vary from each environment, which means that configuration does not have to be hard coded and can be set on each deployment of the code. The variables will be set up in the code base and made available there for use in the application for integrations, logging, etc. (see sample Figure 55).

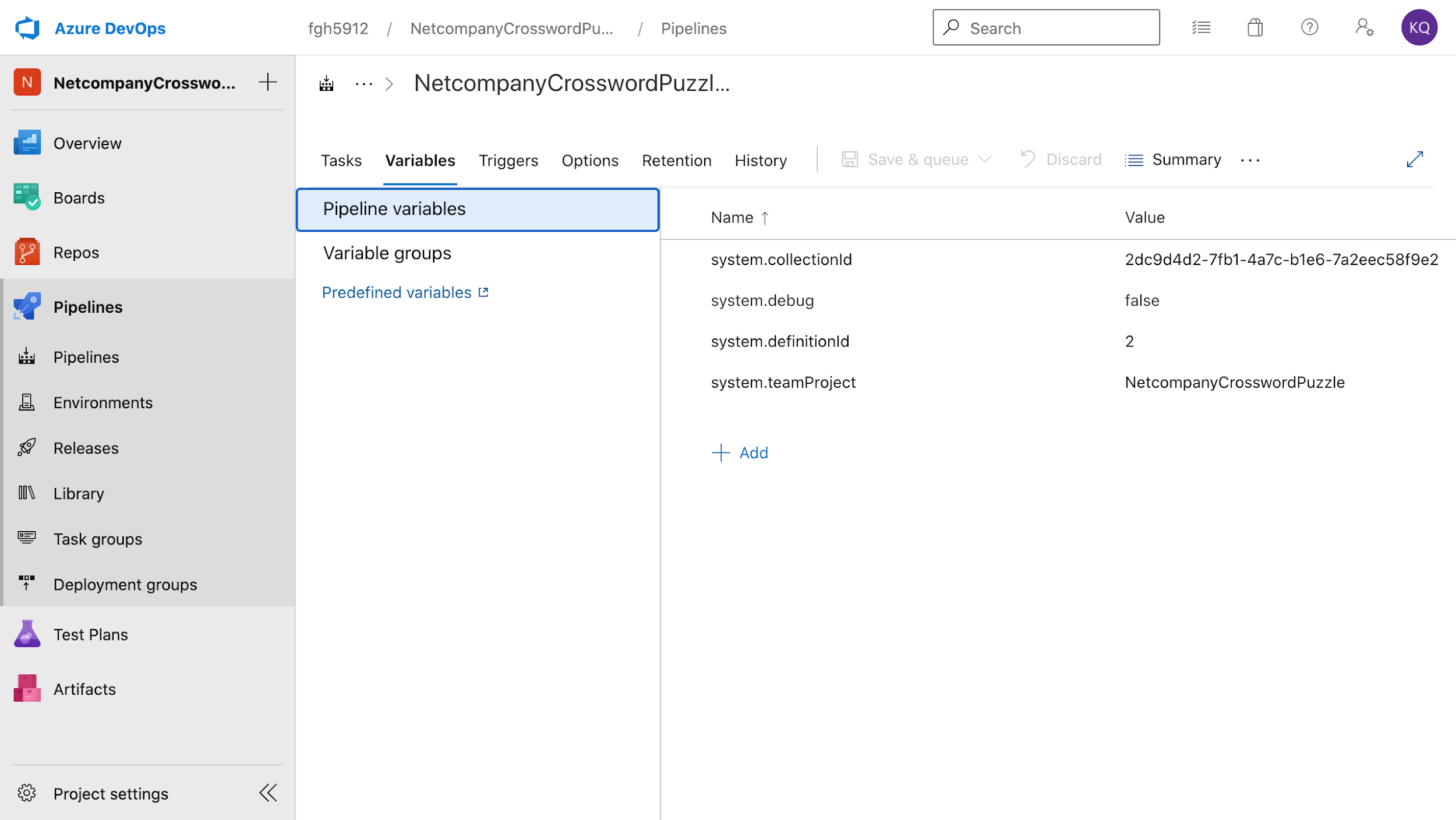


Figure 55 - Pipeline variables overview in Azure DevOps

Azure DevOps also has a security model that allows an administrator to limit access to change control procedures, restrict changes on certain items, setup required reviewers for pull requests or releases and restrict access to source code. The settings can be applied to a group or individual users.

## Performance & Technical Documentation

### Technical Documentation

The Netcompany Methodology includes several deliverables that are an integrated part of the project. The deliverables will be worked on, completed, revisited and revised in the various stages throughout the project. There are three main ones of these covering technical documentation.

#### D0180 – External Interface Design

The purpose of this deliverable is to describe the interfaces and interaction between the solution's own systems and the surrounding external systems. The document must be thought of as a contract between the solution's own systems and the external systems. It is introduced in the design phase, so surprises are kept to a minimum when the development starts.

#### O0400 – Technical Infrastructure

The purpose of this deliverable is to provide a structural overview of the technical infrastructure for the solution. The document goes into details on component diagrams, integrations and the development setup.

#### O0500 – Software Architecture

The purpose of this deliverable is to provide a collective overview of the architecture for the various system stakeholders, including customers, business architects, technical architects, developers, database specialists and operators. The document describes what the system can do, as well as what the system is responsible for.

This document will also include a subsection describing the disaster recovery plan.

### Performance

The performance of the application will in many cases be very much dependent on the performance of the different views and components involved, in particular when the application needs data from various data sources. For example, if loading a page with a particular graph is slow, this is likely the fault of the underlying data source or the un-optimized code.

Performance of the web application on mobile devices is particularly important. If needed, the performance for mobile devices can be improved by using measurements such as making graphs static instead of interactive, or entirely hiding components or graphs if loading times are excessive, for instance in the case of a bad internet connection.

One way to improve performance on the application will be to optimize images. This lets the site load more quickly since traffic on the main servers is reduced.

## Security & Compliance

The requirements for Security and Compliance have been split into two sets; one being the application requirements and the other being hosting requirements.

Netcompany will be the hosting provider of the NCP, which means that some of these requirements are already handled or will be handled by Netcompany’s hosting/operations team.

### Security Measures

The source code of the project won’t be publicly available, as it will be stored in Netcompany’s Azure DevOps repository (see sample Figure 47).

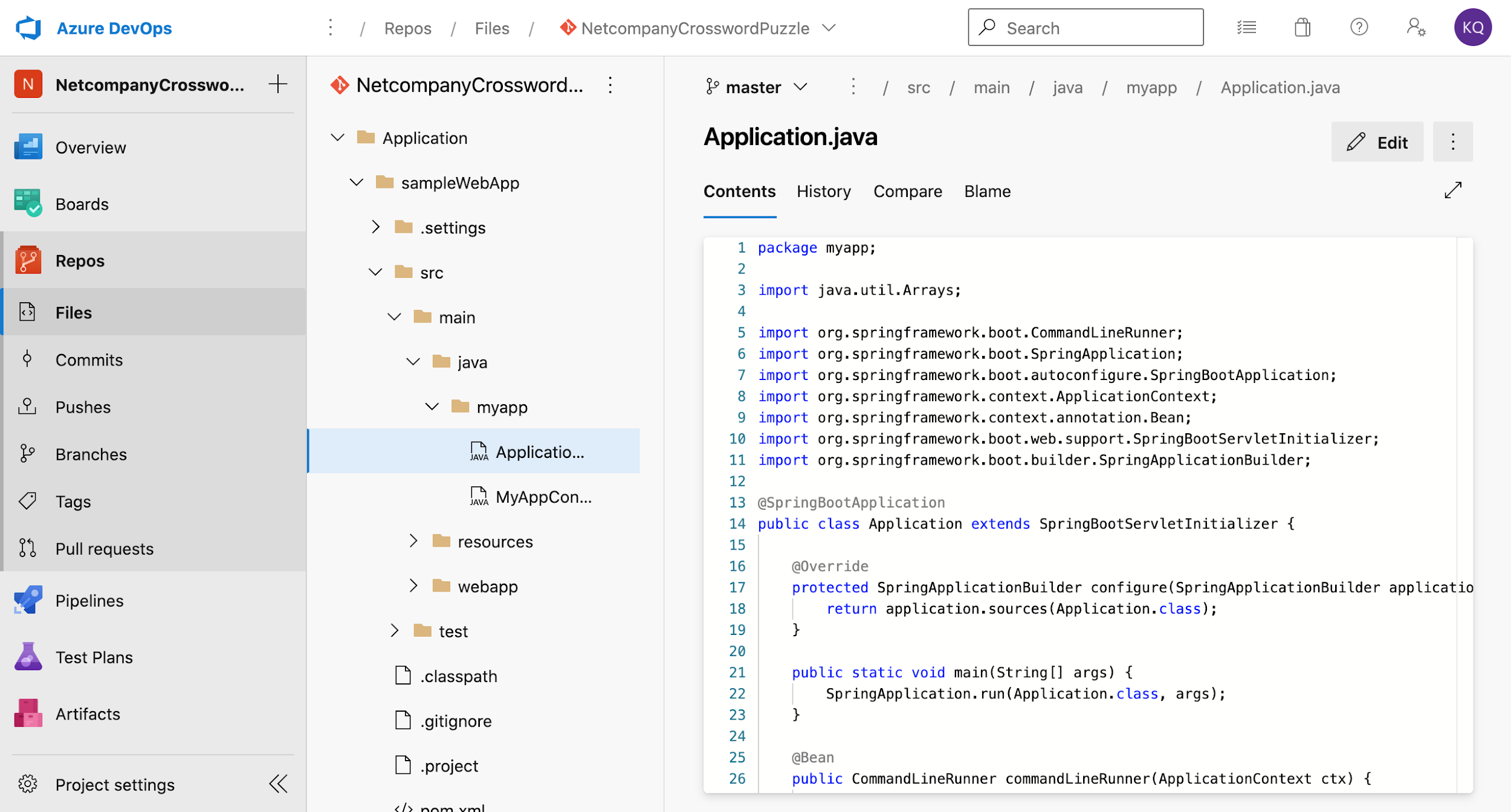


Figure 56 - Azure Devops Repository sample

The application and any associated online services should be using ‘https’ instead of ‘http' for communication, and developers should not carry any remote media or devices with unencrypted data that might compromise the integrity of NCP’s information.

Capacity management will not be implemented on the NCP sites but handled by Microsoft/Netcompany in the Azure tenant hosting the solution after the product is delivered. The system acceptance procedure will be a combination of using the release pipeline for accepting that the code is deployed to the app service running the application, and a sign-off from Netcompany based on the state of the application.

### Authentication

The NCP will utilize simple authentication for the end users since NCP’s scope is small, which is also being used on the corporate website. NCP will be able to provide trust with a customer, if the user’s credential and account exist and not being deactivated in the database. In that way, the user can use their own identities stored in the database. The user will see the login screen from their own browsers. Users without a valid credential will have to be signed up.

No user will be allowed access to any data or services from NCP without being logged in. The customer’s own policies will be applied for reactivation of accounts through the standard functionality in Azure or by the NCP policies. The Azure server supports sending automated emails and blocking users based on risk analysis of the possibility that a user account has been compromised.

## Operation and Hosting

Netcompany will host the application alongside the corporate website in their own on-premise Azure.

The company will also handle the network security to prevent unauthorized network access to the application and any supporting systems in Netcompany’s IT landscape. Netcompany will monitor the production environment for unauthorized processing activities and performance statistics and gain access to Application Insights in the QA environment for inspecting faults and discrepancies that should be solved.

### Encryption of Data

NCP utilizes the Transparent Data Encryption (TDE) of Azure, which will help protect the database. It performs real-time encryption and decryption of the database, associated backups, and transaction log files at rest without requiring changes to the application

In Azure, the default setting for TDE is that the Data Encryption Key (DEK) is protected by a built-in server certificate. The built-in server certificate is unique for each server and the encryption algorithm used is AES 256. All newly created databases in SQL Database are encrypted by default by using service-managed transparent data encryption.

‘Https’ must be used by the web application to prevent attackers from obtaining any information from the application.

### Backup and Restore

The data of NCP should be using the Azure portal's comprehensive backup capabilities. Automated backup must be further configured by Netcompany for the product in a reasonable frequency to be able to restore the app services hosting QA and Production.

# Integration Strategy

The design phase will establish a contract between the solution and surrounding systems to specify, what data should be transferred, and the protocol used. This section will provide an overview of the integrations and the high-level integration strategy for interaction and not specify the business context for each individual integration, as they have been covered in previous sections.

NCP uses Facebook login and Google sign-in for audit logging.

## Glossary of Terms

|  |  |  |
| --- | --- | --- |
| Term | Description | |
| REST API | Representational State Transfer  RESTful services allow other systems to integrate and retrieve or insert data based on a contract.  JSON is a widely used media type for the transition data. |
| SQL | Domain-specific language used in programming and designed for managing data  Held in a relational database management system, or for stream processing in a relational data stream management system.  SQL statements: https://www.w3schools.com/sql/ |

# Technical Infrastructure

This section will provide a structural overview of the technical infrastructure for the solution. The components for the environments are presented from a component overview with descriptions of the logical execution of the components and the relation between them.

## Environments

The major components in the setup for environments and key dependencies are presented in this section. The development environment will be different from the Test & Production environment.

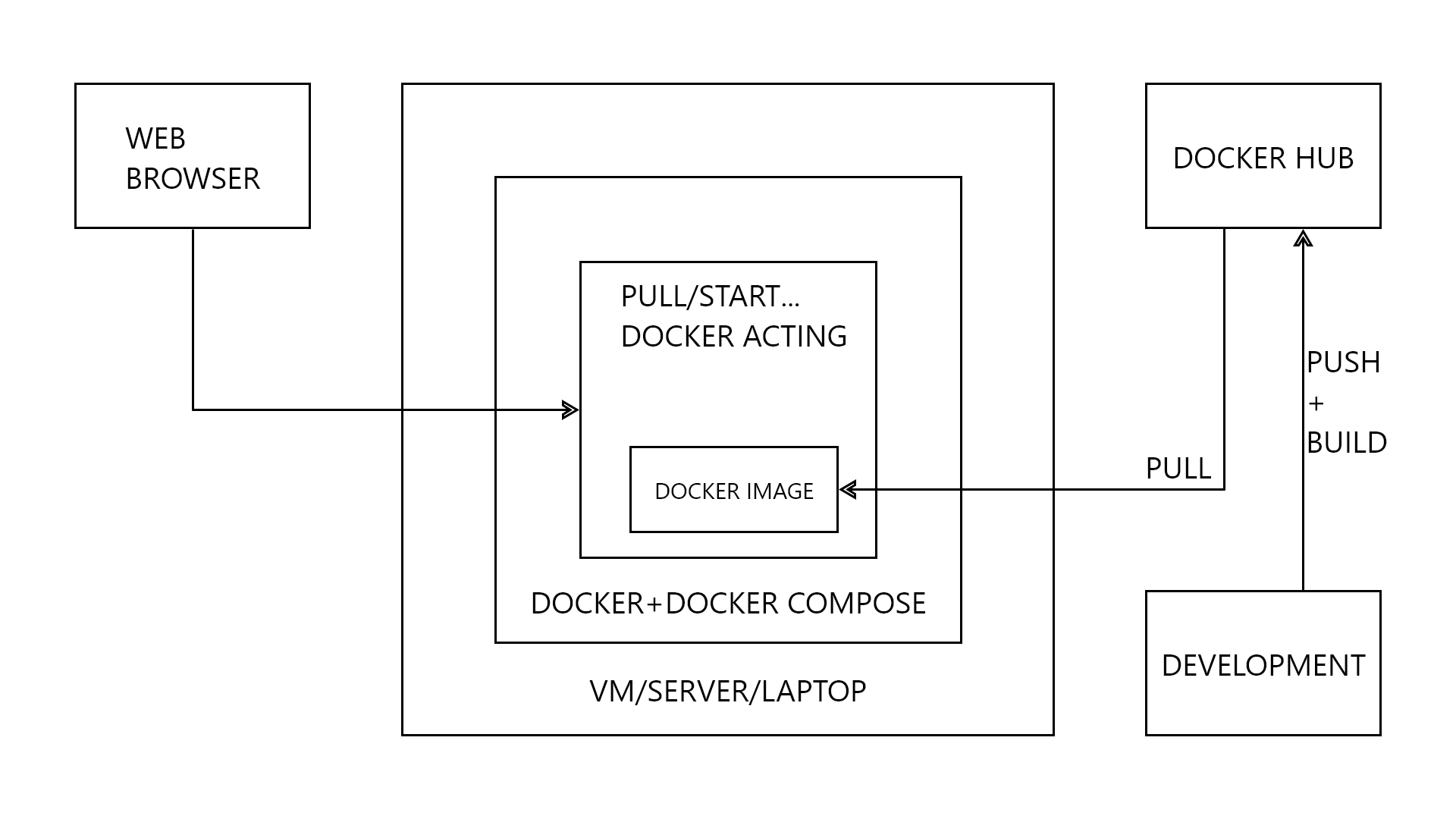


Figure 57 - Environments

### Development

The development environment will be hosted in azure environment with one virtual machine, and using Azure Repo (by importing Github repository) as a repository.

The developer machines and images for building with Docker must be setup with the following applications as minimum (full list will be clarified in the design and build phase):

* Database
  + PostgreSQL database
* JetBrains IntelliJ IDEA
  + The development team will use their own license.
* Microsoft Visual Studio Code (Text Editor)
* NPM (Node Package Manager)
* NodeJS
* ReactJS
* Git
  + Azure Repository
* Google Chrome
* Postman

The developers will login to their own development machine as local administrators and clone the git repository (from either Github or Azure Repo) to their machine (using Git either in the command line or an application to manage git branches) and develop the NCP inside the virtual machine. It is recommended to have a personal development environment for each developer, so the developers do not affect one another.

All commits will go to a new development branch that can be merged into the main branch through Github or Azure DevOps using pull requests and continuous integration.

### Test & Production

The test and production environment will be hosted in a virtual server which will be provided by the Netcompany. It will be setup with the following applications as minimum:

* Docker

Azure DevOps will use the standard integration to deploy the developed code from the code repository to the Production environment. The deployed code will be inside the same app service in Azure with an underlying database for the application. Testing will utilize the Azure Test Plan, which will be broken down into test cases and test steps. Application insights must be enabled and available in the Test environment for Netcompany developers to ensure quality of the application, before it is deployed to production.

The individual details of each component will be clarified in the design phase to ensure that the current technical infrastructure is equipped to support the corporate website.

# Release Plan

Based on the clarifications consolidated within this report, the release plan has been updated. The dates of the milestones are unchanged, the themes of the releases have changed and the scope of the project only has minor adjustments noted to the single requirements.

## Project Roadmap

The project roadmap is established based on priorities of functionality from NCP. The roadmap is still high-level but has placed each theme of functionality into a release.

As established in Project Dependencies , the project is constrained by several dependencies, which is handled by scheduling functionality with dependencies in releases where time permits completion of the dependencies. Due to this, some functionality is planned later than their original priority would demand, in order to accommodate for the dependency. All project and system dependencies have been marked as milestones on Figure 57.

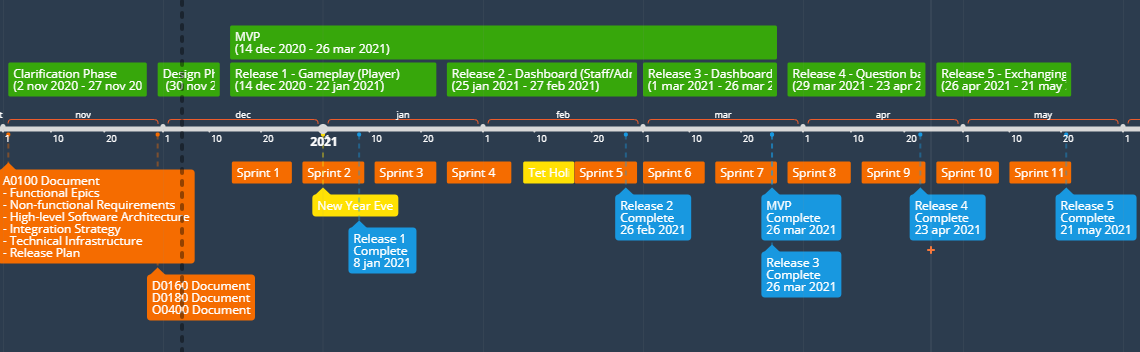


Figure 58 - Project roadmap.

The result is the following releases:

* **Release 1** – Gameplay (Player)
* **Release 2** – Dashboard (Admin/Staff) : Event management
* **Release 3** – Dashboard (Player)
* **Release 4** - Question bank, contributing system and login/logout system
* **Release 5** – Exchanging/Rewarding system and the rest of the Player and Admin Dashboard functionals.

## Changes

### Changes in Requirements.

|  |  |  |
| --- | --- | --- |
| Requirement number | Title | Build estimate (original) |
|  |  |  |

### Change Requests

The current registered change requests can be categorized as either new demands or good ideas that were identified doing the workshops. The changes are thus not required to fulfil the requirements.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Responsible | Estimate (hours) | Status |
|  |  |  |  |  |

## Project Dependencies

### Minimum Value Product(MVP)

Based on the clarification of the functional epics in the previous part of the document, we propose the following MVP:

* Dashboard (Player):
  + Main dashboard
  + View event details
* Gameplay (Player):
  + Waiting room
  + Game room
  + Finished game room
* Dashboard (Admin/Staff):
  + Event Management:
    - Create an event
    - Spectate current event
    - View event details
    - Edit an event

### Project Dependencies

These are the dependencies of the project

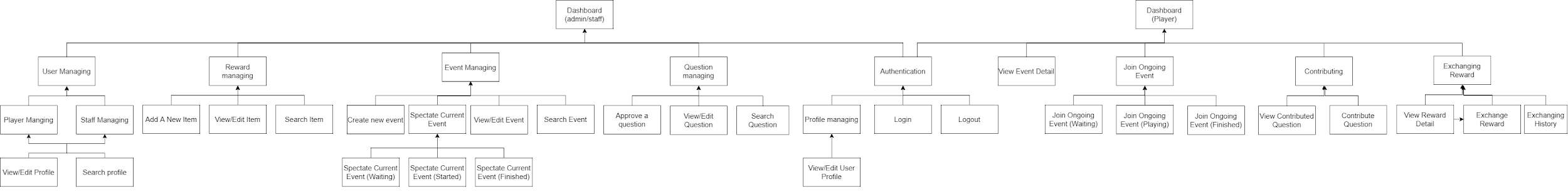


Figure 59 - Project dependencies break down

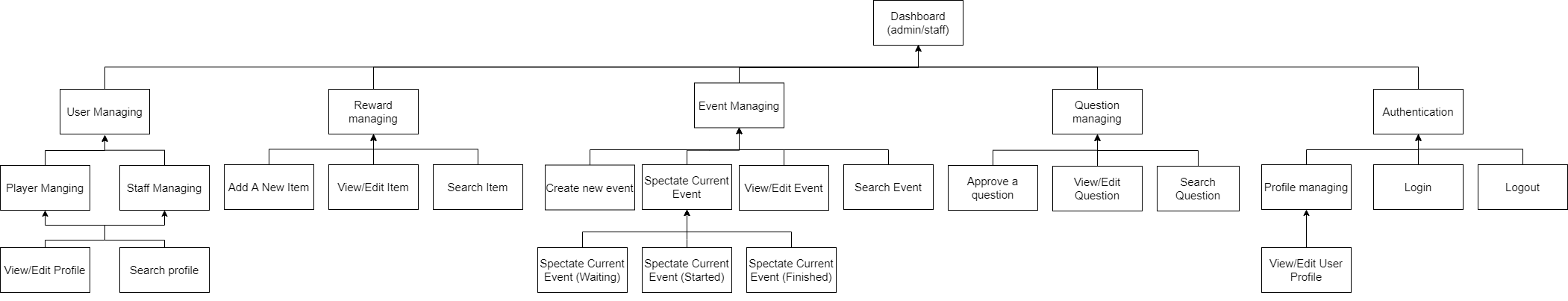


Figure 60 - Project dependencies break down (Staff)



Figure 61 - Project dependencies break down (Player)

## 

## Risks

(Risk này là risk vs góc độ nào?)

The project is actively working with risks management to mitigate materialization of risks impacting the project. Below are the most relevant, active risks identified in the clarification phase.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Impact | Probability | Consequence | Mitigation |
| R1 | Scope and Requirements Mismatch | 4 - High | 3 - Medium | Unrealistic requirements may make the scope of work impossible | Study carefully the requirements and associated scope  Make sure all stakeholders are included in scope development |
| R2 | Scope creep - Uncontrolled scopes | 4 - High | 3 - Medium | Un scope in project's scope may cause some expansion on project schedule | Consider carefully before introducing additional scope change to the project |
| R3 | Skills deficit - Lack of domain knowledge | 3 - Medium | 3 - Medium | Some team members could be incapable of doing some tasks. | Early phase training with new frameworks, technology  Studying and developing simultaneously |
| R4 | Member dropout - Members accidentally drop out of project | 5 - Critical | 1 - Low | Other team members have to double or triple their workload | Reduce the scope |
| R5 | Late on Schedule for release / delivery | 4 - High | 3 - Medium | Some deliverables may not be delivered on time.  Short sprint may cause the same problem as stated above. | Weekly / daily group status report or effective communication to help out team members |

## Outstanding Issues

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Description | Responsible | Priority | Created | Deadline |
|  |  |  |  |  |  |