

Mobile Ticket Checking System Based on Android NFC

Project Report

LI Zhe

Supervisor: **David Rossiter**

2018 Spring

Contents

Abstract.....	3
1. Introduction	4
2. Design.....	5
2.1 System structure.....	5
2.2 Workflow.....	6
2.3 Functions.....	7
2.3.1 Event Management.....	8
2.3.2 Ticket Management.....	9
2.3.3 User Management.....	12
2.3.4 Device Management.....	13
2.4 Database	13
2.4.1 Entities.....	13
2.4.2 Table structure.....	14
3. Implementation	17
3.1 Mobile client	17
3.1.1 Layout.....	17
3.1.2 RFID tag reading.....	18
3.1.3 Device token and event ID configuration	18
3.1.4 Ticket issuance.....	19
3.1.5 Ticket checking.....	19
3.1.6 Clear screen.....	20
3.2 Web client.....	21
3.2.1 Layout.....	21
3.2.2 Pages.....	21
3.3 Server	23
4. Conclusion.....	25
5. Appendix.....	26
5.1 Minutes for the first meeting	26
5.2 Minutes for the second meeting.....	26
5.3 Minutes for the third meeting	27
5.4 Minutes for the fourth meeting.....	27
6. References.....	29

Abstract

Nowadays electronic tickets are wildly used in admission control system. In traditional electronic ticket checking systems, each physical ticket is linked to one electronic ticket record. Ticket checking devices will get ticket identification ID from physical tickets then check the related ticket records.

This project aims at building a new system for ticket checking. In this system, there will be no customized physical tickets and ticket checking terminals. Ordinary RFID tags, for example, transportation cards or campus cards are used as identification labels of tickets and mobile phones which supports NFC function will act as ticket checking devices. This brings a light and green solution to electronic tickets management.

1. Introduction

Management of admission tickets can be an important part in organization of public activities. There are several key factors to evaluate the performance of one ticket management system:

Security: To prevent forgery, illegal alteration and non-authorized transfer of tickets.

Speed: To reduce time cost of issuing and checking of tickets.

Flexibility: The system should support wild range of tickets and perform well in different environments.

Eco-friendly: To reduce waste and pollution during ticket issuing and checking.

Core content of this project's work is to build a new generation ticket checking system. In this system, there are no physical tickets. All tickets are electronic records stored in system's database, and each ticket record contains one special field: tag ID. Tag ID is the unique identity number of an RFID tag, and it can be read by mobile phones which support NFC function. There are different kinds of RFID tags in our daily life, such as transportation card, campus card, metro tickets and citizen identity card. Even some credit card which supports contactless payment can also act as RFID tag in this system. In this way, tickets are linked with RFID tags, ticket issuance and checking can be done by reading the tags.

2. Design

2.1 System structure

This system contains one server and several clients. There are two kinds of clients: mobile client and web client. Mobile client is the mobile phone which installed the ticket checking application, web client is the web platform of this system.

Structure of this system is shown by figure 1.

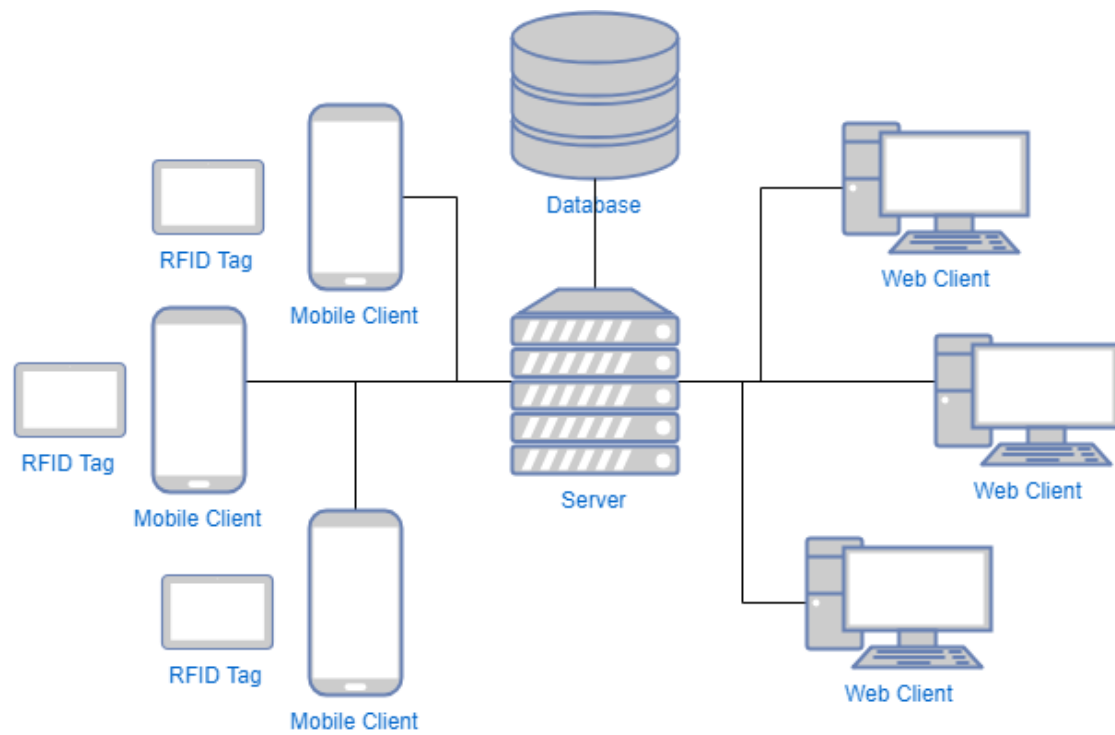


Figure 1 System Structure

2.2 Workflow

Standard workflow of this system is:

1. Create an event. (via web client)

Add an event on add event page of web client.

2. Issue tickets to event (via web client or mobile client)

This can be done in two ways: by entering RFID tag ID and event ID on web client page or scanning RFID tag with mobile client device.

3. Check tickets (via mobile client)

Scan RFID tag with mobile client device to get ticket information.

Otherwise, after creating an event, user may also need to search or update events information, same for tickets. To use mobile clients, user needs to configure device information in the system first.

A sample workflow of this system is shown as figure 2.

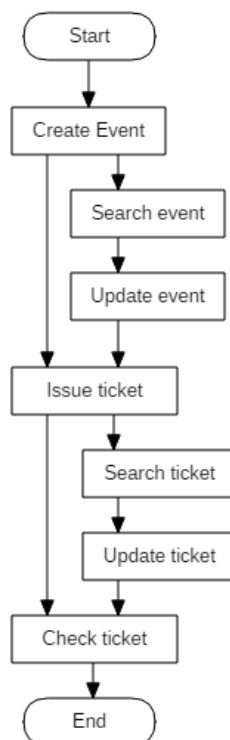


Figure 2 A Sample Workflow

2.3 Functions

Functions of this system can be divided in 4 modules: ticket management, event management, user management and device management. Ticket checking and issuing are main functions of system and put under 'ticket management' module.

Use case diagram is shown as figure 3.

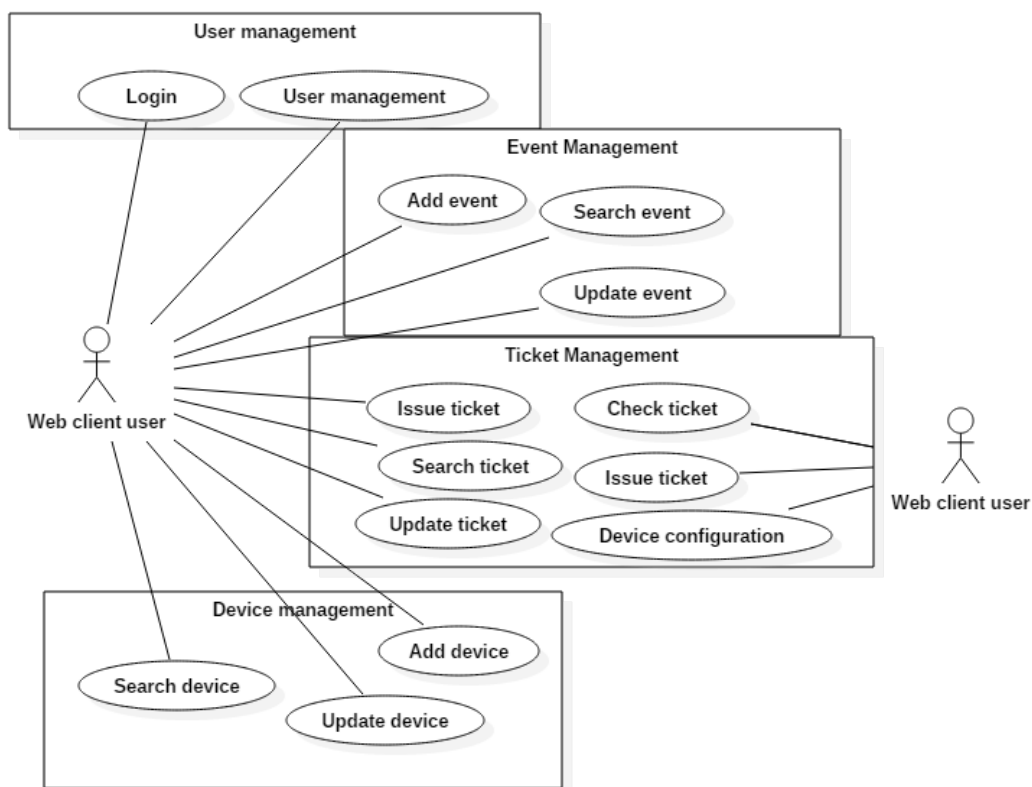


Figure 3 Use Case Diagram

2.3.1 Event Management

Event management functions are provided on web client only.

Add event

Purpose: Add one event to system for following ticket management

Precondition: User already logged in system.

Flow of events:

1. User directs to 'Add an event' page
2. Fill the event information on page.
3. Click 'Add' button to add this event to system.

Result: If event is added successfully, there should be one prompt shown on page.

Exceptions: On step 1, if user's login status expires, page will turn to login page automatically. On step 2, if user does not fill all required blanks, he can not click 'Add' Button. After step 3, if failed to add the event, one failure prompt will appear on page.

Search event

Purpose: Search one event for further operation

Precondition: User already logged in system.

Flow of events:

1. User directs to 'All events' or 'Active Events' page.
2. All events shown on page, user can filter the events on page.

Exceptions: If user's login status expired, page will directs to 'Login' page.

Update event

Purpose: Update information of one event.

Precondition: User is logged in and already got ID of event to update.

Flow of events:

1. User directs to 'Update an event' page.
2. Enter the ID of event to update, click 'Load event' button.
3. Browse and update the information of event.
4. Click 'Update' button

Exceptions: On step 1, if user's login expired, page will direct to 'Login' page. On step 2, if ID input does not exist, there will be a prompt on the page. On step 3, if there are any failure in database updating or web communication, an error prompt will appear on the page.

2.3.2 Ticket Management

Ticket management functions are provided on both web client and mobile client.

Check ticket

Purpose: Check the ticket related to one RFID tag.

Precondition: The mobile device is configured with valid device token and event ID.

Flow of events:

1. Set device to 'ticket checking' mode.
2. Put the RFID tag to NFC sense area of device.
3. Device reads ID from tag, send tag ID and event ID to server.
4. Server finds the related ticket, modify ticket status and return the result.
5. Device display ticket information and checking result on screen.
6. Clear the screen for next ticket.

Exception: On step 2, if RFID tag is not recognized by device, there will be one warning prompt. On step 5, if failed to check the ticket or fetch the information, there will be one error prompt.

To make a better understand of this progress, sequence diagram of ticket check function is shown as figure 4.

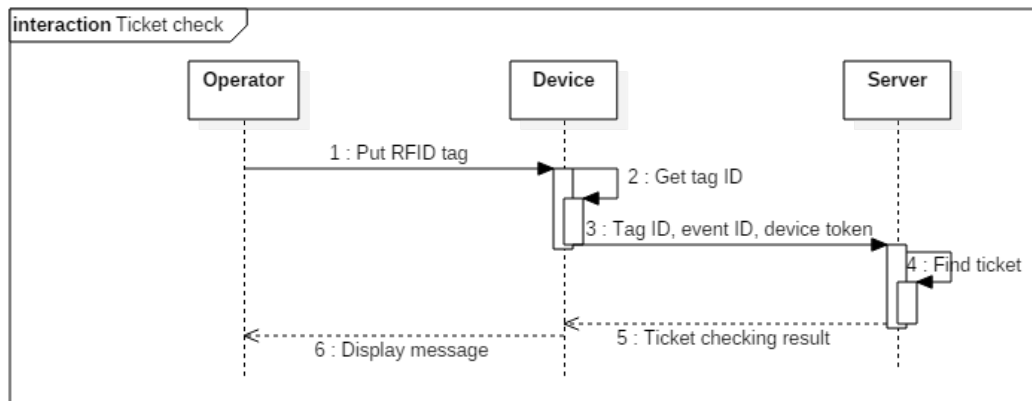


Figure 4 Check Ticket

Issue ticket

Purpose: Issue one ticket to RFID tag using mobile client.

Preconditions: Mobile device is configured with valid device token and event ID.

Flow of events:

1. Set device to 'ticket issuance' mode.
2. Put the RFID tag to NFC sense area of device.
3. Device reads tag and get tag ID, transmits required information to server.
4. Server issue tickets according to data sent by device, return the result
5. Device display the result.
6. Clear screen for next ticket.

Exceptions: On step 2, failed ticket scanning will cause one warning prompt. On step 5, if failed to issue ticket or fetch information from server, an error prompt will appear on the screen.

Note: Ticket issuing operation on web side is very similar to event adding.

To make a better understand of this function, sequence diagram is shown as figure 5.

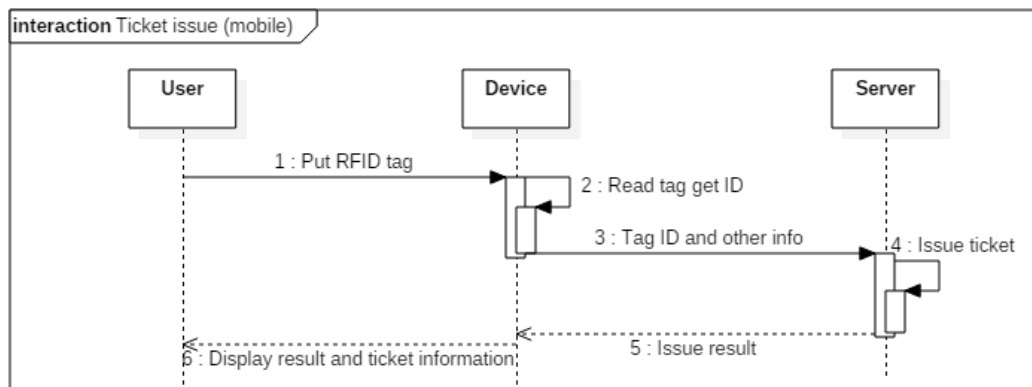


Figure 5 Issue Ticket on Mobile Client

Device configuration

Purpose: Configure the device with device token and event ID, similar to ‘Log in’ operation.

Precondition: mobile client application is installed to device, and device can connect to network.

Flow of events:

1. User open the application
2. Click ‘Admin’ button, input the device token and event ID, then click ‘Confirm’ button.
3. Device name and event name is shown on screen, ticket issue or check mode is activated.

Exceptions: On step 3, if device token or event ID are invalid, an error prompt will appear on screen and can not continue to ticket issue or check progress.

Sequence diagram of this function is provided to make a better understand and shown as figure 6.

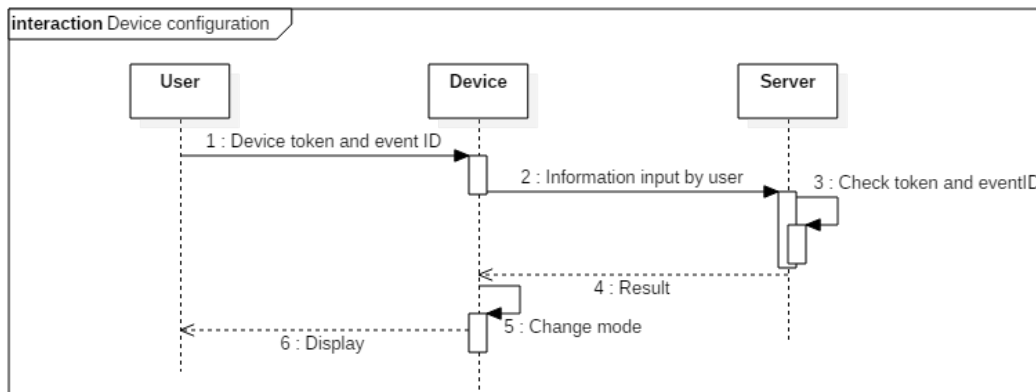


Figure 6 Device Configuration

2.3.3 User Management

User management functions are provided on web client only.

User login

Purpose: Allow user login to the ticket management system on web client

Precondition: Computer connect to network and proper browser

Flow of events:

1. User direct to login page.
2. Enter user login name and password
3. Click 'Login' button.

Result: If succeed to login, a prompt will appear and page directs to index page.

Exceptions: On step 2, if login name or password are not filled, 'Login' button will be disabled. On step 3, if failed to login or fetch information from server, an error prompt will appear.

User management

Purpose: Allow user modify password or login name.

2.3.4 Device Management

Device management functions are provided on web client only, and they are very similar to event management.

2.4 Database

2.4.1 Entities

There are 5 entities in this system: ticket, event, user, user's token and device.

Ticket: Each ticket entity is equal to one physical ticket, it only belongs to one specified event and RFID tag.

Event: Each event entity is equal to certain event in reality (for example, concert, movie, public activity, exhibition and so on). One event can have many tickets, but one RFID tag can only be linked to one ticket in one event.

User: User of web client.

User's token: When user login on web client, system generates one token for the user and store it into browser's cookie.

Device: Mobile device which acts as mobile client in system. Each device contains one unique ID and login token. Mobile client should configure the device ID and token in mobile client application then connect to the system.

E-R diagram is shown as figure 7.

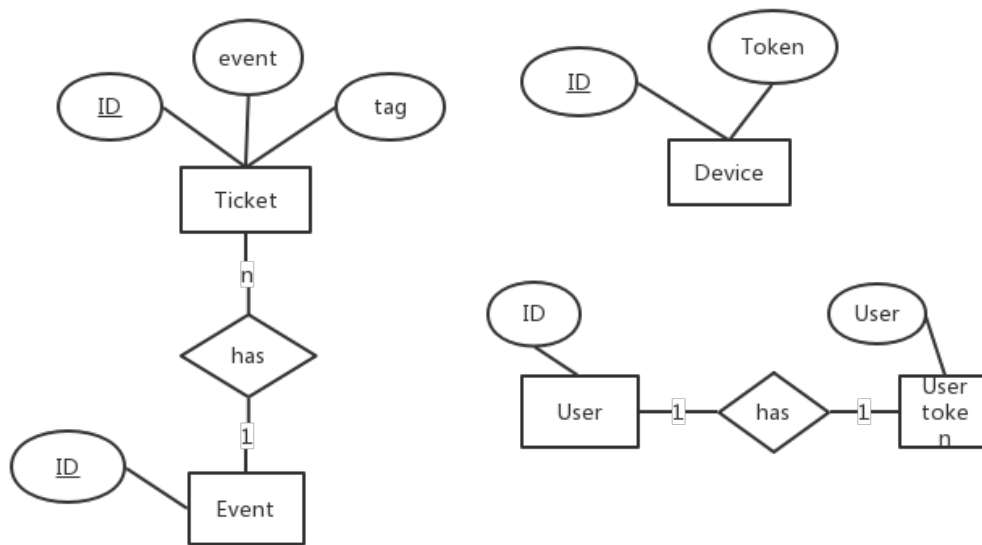


Figure 7 E-R Diagram

2.4.2 Table structure

Structure of data tables of system is shown by Table 1 ~ 5.

Table name: tc_event Purpose: To store events data.

Table 1 Event Record

Column	Type	Description
tc_event_id	int	Primary key, unique ID of event
tc_event_name	varchar	Name or title of one event
tc_event_starttime	int	Start time of event, Linux timestamp
tc_event_endtime	int	End time of event, Linux timestamp
tc_event_state	int	0: Open or Active 1: Closed or Inactive
tc_event_ticketlimit	int	Ticket number limit of this event
tc_event_creator	int	Creator of this event, FK: tc_user_id
tc_event_state will be updated manually or by crontab		

Table name: tc_ticket Purpose: To store tickets data

Table 2 Ticket Record

Column	Type	Description
tc_ticket_id	int	Primark key, unique ID of ticket record
tc_ticket_ticketref	varchar	Reference number for external use
tc_ticket_tagid	varchar	ID of RFID tag linked to this ticket
tc_ticket_eventid	int	ID of event this ticket belongs to, FK tc_event_id
tc_ticket_issuetime	int	Time of issuing this ticket, Linux timestamp
tc_ticket_issuer	int	Issuer's user ID or device ID
tc_ticket_issuertype	int	0: Issued by web client 1:Issued by mobile client
tc_ticket_checktime	int	Time of checking this ticket, Linux timestamp
tc_ticket_checker	int	Check of this ticket, FK tc_device_id
tc_ticket_type	int	Type of this ticket, 0=Ordinary, 1=Special
tc_ticket_state	int	Status of this ticket, 0=open for use, 1=used
tc_ticket_note	varchar	Any notes for this ticket
Unique(tc_ticket_tagid, tc_ticket_eventid) to keep one tag can only be linked to one ticket in one event.		

Table name: tc_user Purpose: To store users data

Table 3 User Record

Column	Type	Description
tc_user_id	int	PK, unique ID of one user
tc_user_loginname	varchar	Name for login use, can modify by user
tc_user_showname	varchar	Shown name for user, can modify by user
tc_user_password	varchar	Password after hash
tc_user_type	int	Type of user, 0=Normal 1=Super
tc_user_state	int	Status of user, 0=Open 1=Closed
Unique(tc_user_loginname) to avoid two same login name user.		

Table name: tc_usertoken Purpose: To store users' tokens

Table 4 User Token Record

Column	Type	Description
tc_usertoken_uid	int	FK: tc_user_id
tc_usertoken_token	varchar	Token to identify real user
tc_usertoken_timelimit	int	Expire time of this token
Unique(tc_usertoken_uid) One user can only have one valid token.		

Table name: tc_device Purpose: To store devices' tokens.

Table 5 Device Record

Column	Type	Description
tc_device_id	int	PK, unique ID of one device
tc_device_name	varchar	Name of device, will show on screen
tc_device_token	varchar	Token of this device
tc_device_tokenexpire	int	Expire time of token
One device can only have one token at one time.		

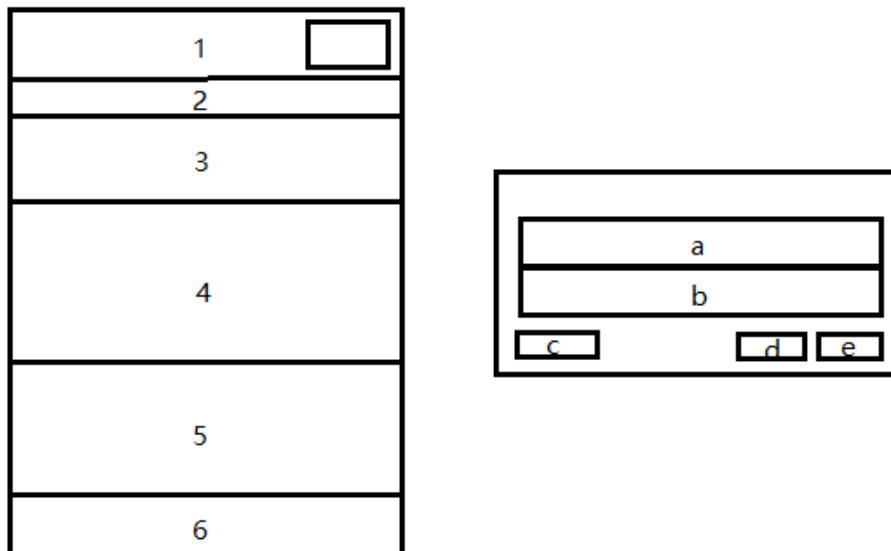
3. Implementation

3.1 Mobile client

On mobile client side, an Android application is developed. Any mobile device supports NFC and runs on Android can install this application then act as one mobile client device in this system.

3.1.1 Layout

The layout contains two parts: main activity and configuration dialog, as shows.



Left side: main activity.

- 1: Device name and configuration button.
- 2: Current mode, ticket issue mode or check mode.
- 3: Event title.
- 4: Ticket check or issue result
- 5: Tag and ticket information.
- 6: Clear screen button.

Right side: configuration dialog, triggered by click configuration button on main activity,

- a: Fill event ID

- b: Fill device token.
- c: Change mode button
- d: Cancel button
- e: Confirm button

3.1.2 RFID tag reading

Device should have NFC function and give permission to this application.

After application started, we put one RFID tag to NFC sense area of device, the unique ID of RFID tag will be read and displayed on screen.

If device token and event is not configured, there will be one prompt telling user that the device can not check or issue ticket now.

Figure 8 shows the RFID tag ID reading result and a warning prompt.

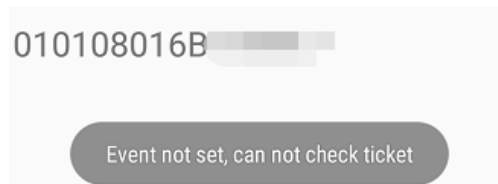


Figure 8 RFID Tag ID and prompt

3.1.3 Device token and event ID configuration

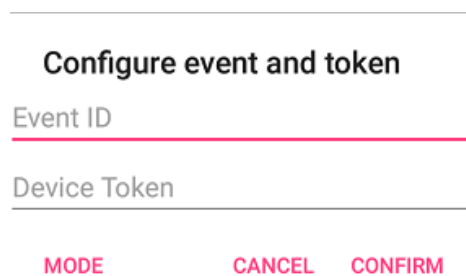
Click 'Admin' button, configure dialog will appear. After enter the event ID and device token, click 'Confirm'. application will sent the data input to server.

If both event ID and token are valid, application will get device name and event title from server then display it on main activity, and mode is set to ticket checking automatically.

If there are any error in the information user input, application will get error information from server and display one prompt about failed configuration. At this time device can not do any ticket operation until correct event ID and device token are configured.

Click 'mode' button on configuration dialog, will change the mode. Default mode after application started is ticket check mode.

Configuration dialog is shown in figure 9.



The image shows a configuration dialog box titled "Configure event and token". It contains two text input fields: "Event ID" and "Device Token". Below the input fields are three buttons: "MODE", "CANCEL", and "CONFIRM". The "MODE" button is highlighted in red.

Figure 9 Configuration Dialog

3.1.4 Ticket issuance

If event ID and device token are configured, and change mode to 'Issuance mode', put the RFID tag to NFC sense area, client gets tag ID then sent the information to server.

If one ticket is successfully issued, the ticket information will be returned and displayed on screen.

3.1.5 Ticket checking

When device is set to 'Ordinary mode', put RFID tag to NFC sense area, the mobile client gets tag ID then sent the information to server.

If there were one available ticket for current event linked to this tag, server will return the information of this ticket with success code, and the result for ticket checking will be displayed on screen.

One sample screenshot of ticket issuing and ticket checking is shown as figure 10.

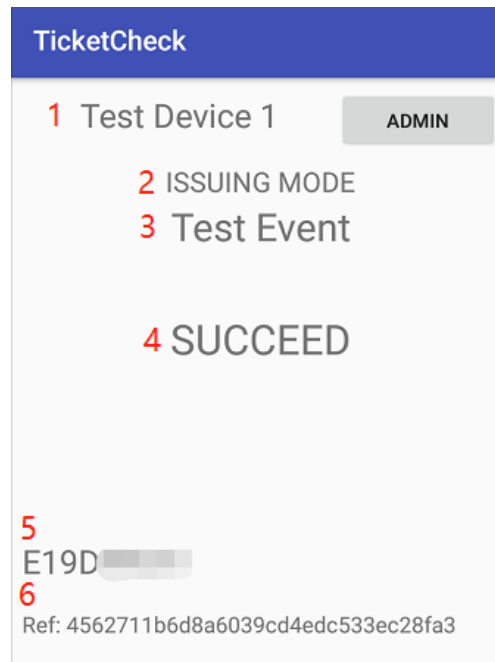


Figure 10 Sample Screenshot

1. Name of device 2. Current mode, ordinary mode(checking) or issue mode(issuing) 3. Event name 4. Status, standby or succeed or failed 5. RFID tag ID read 6. Ticket reference number which issued or checked for current operation

3.1.6 Clear screen

Click 'Clear' button, ticket information shown on current screen will be cleared.

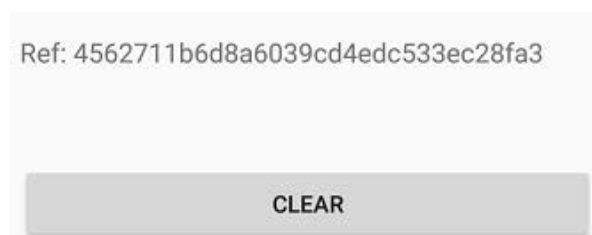
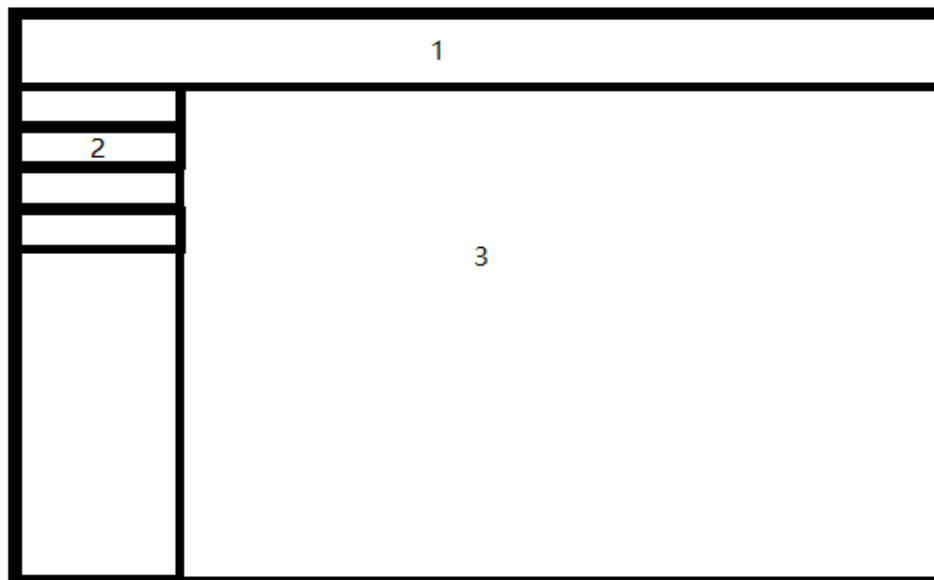


Figure 11 Clear Button

3.2 Web client

3.2.1 Layout

Layout of most pages is shown in fig.



Three main sections:

1. Title of system
2. Menu
3. Information and operation area.

3.2.2 Pages

Pages for event management contains events list page, active events list page, event adding page and event updating page.

Pages for event management contains: event list page, active events list page, event adding page and event updating page.

Pages for ticket management contains: ticket issuance page, ticket searching by tag ID page, ticket searching by event page, ticket updating page.

Pages for device management contains device list page and device updating page.

Figure 12~14 are sample page screenshots for event list page, ticket searching by event page and ticket updating page.

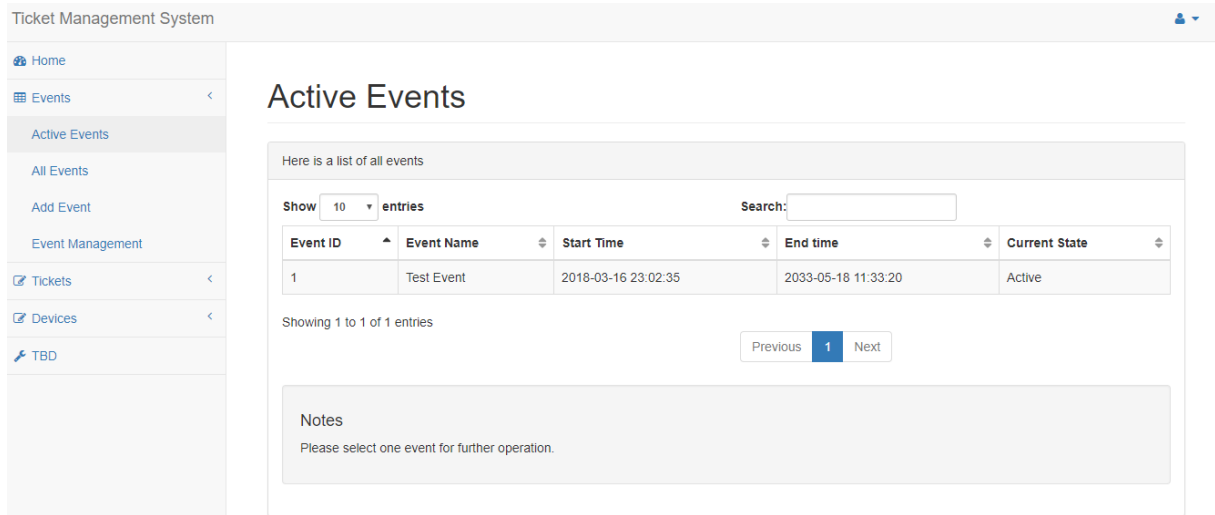


Figure 12 Event List page

Find Ticket by Event

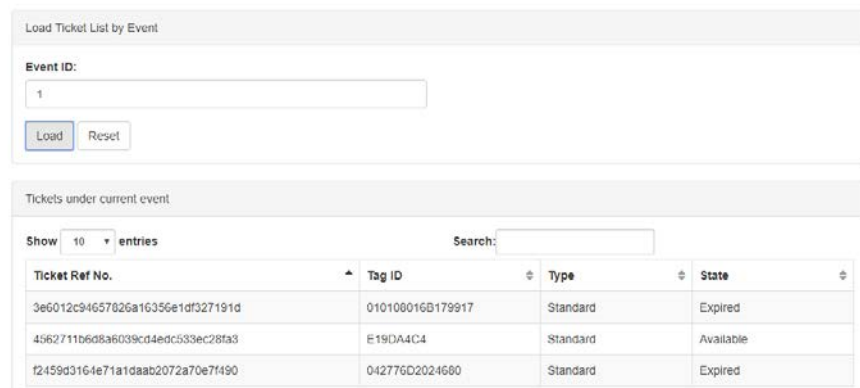


Figure 13 Ticket Searching Page

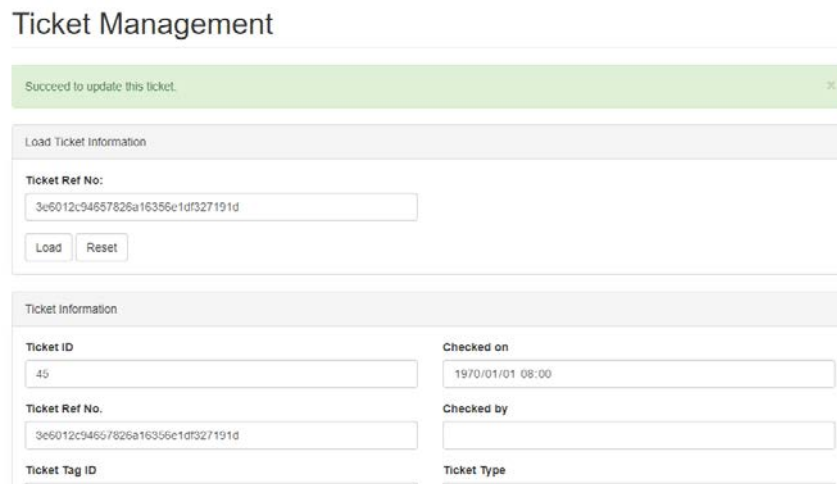


Figure 14 Ticket Updating Success page

3.3 Server

Both mobile clients and web clients communicate with server by API.

For mobile clients, mobile application will push POST requests using OKHttp library to server and get the return value, parse it then display any results on screen.

For web clients, the page will communicate with server using Ajax.

Figure 15 and 16 showed one test done for one API to get ticket formation by ticket reference number. This API is called on ticket searching by reference page by clicking 'Load' button, Ajax is used to change the content on page. Ticket information is returned in JSON format, the page will extract and display it.

POST http://54.65.104.█/tc/logic/getTicketbyref.php

Edit and Retry

200 OK 318 bytes 341 ms

[View Request](#) [View Response](#)

HEADERS

Accept: */*
Accept-Encoding: gzip, deflate
Content-Length: 102
Content-Type: application/x-www-form-urlencoded
User-Agent: runscope/0.1

PARAMETERS

ticketRef: 3e6012c94657826a16356e1df327191d
userID: █
userToken: █

BODY

userID=█;ticketRef=3e6012c94657826a16356e1df327191d&userToken=█

[view raw](#)

Figure 15 Request for one API test

POST http://54.65.104.█/tc/logic/getTicketbyref.php

Edit and Retry

200 OK 318 bytes 341 ms

[View Request](#) [View Response](#)

HEADERS

Content-Length: 318
Content-Type: text/html; charset=UTF-8
Date: Sat, 12 May 2018 09:16:28 GMT
Server: Apache/2.4.27 (Amazon) PHP/7.0.29
X-Powered-By: PHP/7.0.29

BODY

```
{ "type": "getTicketResult", "state": 200, "ticketID": "45", "ticketRef": "3e6012c94657826a16356e1df327191d", "ticketTag": "0101080166179917", "ticketEvent": "1", "
```

[view raw](#)

Figure 16 Response for one API test

4. Conclusion

Following functions are implemented in this project:

- Ticket issuing on mobile client side and web client side.
- Ticket checking on mobile client side.
- Creating, searching and updating for events and devices.
- User management for client side.

Average time spent for single ticket issuing or checking action on mobile device is 0.4 second. Compared with manual operation (around 90 tickets per minute), ticket checking by mobile device can save time and manpower cost to a certain extent.

There are still many areas to improve in this project.

On mobile client side, status icons and timer can be added on layout. Functions such as ticket information querying, ticket operation history querying can be added. Stress test of system can be performed for improving the availability and stability of the system.

On web client side, HTTPS connection needs to be used to keep data safe. More input validation on page should be added to keep data clean. A statistic page can be added for user to analysis the ticket issuing and checking progress.

On server side, more security measures should be taken to prevent the system from being attacked such as DDoS. More types of tickets and users need to be supported. The database needs to be optimized to improve querying and updating efficiency

5. Appendix

5.1 Minutes for the first meeting

Date: 2018/02/26

Time: 11:00

Place: Room 3554

Attending: Li Zhe, Prof. David Rossiter

Absent: None

Recorder: Li Zhe

Approval of minutes:

No minutes to approve.

Discussion:

Brief introduction of project, system structure and layout design.

Work plan and progress monitor tools.

Meeting adjournment and next meeting:

Meeting adjourned at 11:15. Next meeting: around Mar. 19th.

5.2 Minutes for the second meeting

Date: 2018/03/19

Time: 11:00

Place: Room 3554

Attending: Li Zhe, Prof. David Rossiter

Absent: None

Recorder: Li Zhe

Approval of minutes:

Minutes for the first meeting approved, no amendment required.

Discussion:

Work progress on server side and mobile client side displayed.

Layout of mobile client needs to be optimized for easier to read.

Design of web client, Ajax should be used to update the content of page without refreshing.

Meeting adjournment and next meeting:

Meeting adjourned at 11:15. Next meeting: around Apr. 9th.

5.3 Minutes for the third meeting

Date: 2018/04/09

Time: 11:00

Place: Room 3554

Attending: Li Zhe, Prof. David Rossiter

Absent: None

Recorder: Li Zhe

Approval of minutes:

Minutes for the second meeting approved, no amendment required.

Discussion:

Table design and query optimization.

Adjust the layout of mobile client to make the distinction between modules clearer.

Status code should be used in response package of server to make it easier for mobile client to judge the result of ticket operation.

Meeting adjournment and next meeting:

Meeting adjourned at 11:15. Next meeting: around May. 9th.

5.4 Minutes for the fourth meeting

Date: 2018/05/09

Time: 11:00

Place: Room 3554

Attending: Li Zhe, Prof. David Rossiter

Absent: None

Recorder: Li Zhe

Approval of minutes:

Minutes for the third meeting approved, no amendment required.

Discussion:

Certain areas improved: Prevention of mis-operation added to mobile client, form validation added on several pages of web client.

Reviewed the draft of report and talked about revising suggestions.

Meeting adjournment and next meeting:

Meeting adjourned at 11:25. Final meeting.

6. References

https://en.wikipedia.org/wiki/Radio-frequency_identification

https://en.wikipedia.org/wiki/Near-field_communication

<https://developer.android.com/guide/topics/connectivity/nfc/>

https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX/Getting_Started

<http://square.github.io/okhttp/>

<https://jquery.com/>

<http://www.php.net/>

<https://www.mysql.com/>

<https://medoo.in/>