|  |  |  |
| --- | --- | --- |
| ASE Group Project - Documentation | December 12  2012 | |
|  | |  |

Group 4:

Sean Chapman: Candidate Number: 102223

Jolanta Rusecka: Candidate Number: 102150

Chris Walker: Candidate Number: 102227

[Andria Orphanidou](https://plus.google.com/u/0/101226752234022848438?prsrc=4): Candidate Number: 102148

Ewen Cluley: Candidate Number: 102230

Contents

[1. Planning products 2](#_Toc341213960)

[2. Requirements documentation 2](#_Toc341213961)

[2.1 Requirements document 2](#_Toc341213962)

[Stowaways overview 2](#_Toc341213963)

[Stakeholders 2](#_Toc341213964)

[3. Use case model 5](#_Toc341213965)

[Events table 5](#_Toc341213966)

[Crud matrix 6](#_Toc341213967)

[Use case diagram 7](#_Toc341213968)

[List of actors and descriptions 8](#_Toc341213969)

[Individual use case diagrams 9](#_Toc341213970)

[Use case: Add a boat 9](#_Toc341213971)

[Use case: Check boat availability 10](#_Toc341213972)

[Use case: Make a payment 11](#_Toc341213973)

[Use case: Make a booking 12](#_Toc341213974)

[Use case descriptions 13](#_Toc341213975)

[4. E-R model 14](#_Toc341213976)

[5. Test plan and test cases 18](#_Toc341213977)

[Test plan 18](#_Toc341213978)

[6. Class diagram 21](#_Toc341213979)

[Appendices 22](#_Toc341213980)

# Document overview

The purpose of this report is to document the following:

* overview of the whole project, and its organisation
* summarise development life cycle approach taken
* modeling approach taken (design patterns, choose the abstractions)
* the structural design through the use of class diagram
* user-system interaction via use-case diagram and use case descriptions
* black box testing – ( user interface testing )

# Project overview

Use GitHub –version controlling tool

Develop Android app – eclipse environment

Stakeholders

The main stakeholders are the project team members, product owner and end users.

# Planning products

# Requirements documentation

Android App – Clock and GPS location display

Extension 1

Extend the App you have already produced so that periodically it signs in to an internet service with its location

Extension 2

Extend your application so that the user of the App can "check in" to particular locations. They should then be able to see when other users of the app have visited the same location (or, possibly competing locations of the same type....??), see other users' ratings and/or reviews and enter ratings/reviews themselves.

Extension 3

The requirements are grouped into following categories:

* Functional
* Non-functional requirements
* User requirements
  1. **Functional requirements**

The new system must be able to allow to:

1. The system should allow user registration and verification.
2. The system should grant [access](http://www.modernanalyst.com/Community/Forums/tabid/76/forumid/17/postid/4092/scope/posts/Default.aspx) to the user after he provides username and password.
3. The system should remember the user credentials, if user specifies wishes to do so.
4. The system should display periodically updates gps location, and current time.
5. The system should provide read access of the following: other user comments, places description.
6. The system should allow the user to create comments and add new places to the system.
7. The system should allow the user to check into places.
   1. **Non-functional requirements**

The system must be able to implement functional requirements in a following way:

1. Interoperability. Aapplication allowing concurrent users login. Application works the same on different versions of Android operating systems and various Android devices (tablets, phones).
2. Usability. Ease of use and user friendly design.
3. Reliability. The consistent performance of the system. For example: All buttons support their required performance.
4. Scalability. It is possible to development additional functionalities and system can cope with increased volume of user’s without its performance being affected.
5. Secure system. Access restrictions. Unique Login details required, user verification required.
6. No inconsistent or repeated data allowed.
7. No data loss allowed.
   1. **User requirements (Backlog)**

**In order to gather additional user requirements these have been collated in the Backlog:**

…..

## 3. Use case model

1. Register new user
2. User log in
3. Display GPS location
4. Display Time
5. Periodically update user location
6. Check into places
7. Display places in X radius
8. Display other user location in X radius
9. Add a place description
10. Add a comment
11. Display comments

### Crud matrix (TBC)

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Entity**  **Use case** | **Client** | **Booking** | **Boat** | **Pmnt** | **Mnt** | CM | Class | Boat  Week | Boat Yard | Weeks | Price |
| Create seasonal calendar |  |  |  |  | U |  | R | C |  | C | R |
| Check Boat Availability | C | R | R |  | R |  | R | R |  | R |  |
| Make a booking | C | CU | R |  | R | C | R | U |  | R | R |
| Make payment | R | RU |  | C |  |  |  | U |  |  |  |
| Amend booking | R | CRU |  | R |  |  |  | RU |  | R |  |
| Cancel booking | R | RD | U | UD |  | R |  | RUD |  |  |  |
| Generate invoice | R | RU |  | R |  |  |  | R |  |  |  |
| Assign special discount | R | RU |  | C |  |  |  |  |  |  |  |
| Produce weekly reports |  | R |  | R |  |  |  |  |  | R |  |
| Schedule maintenance work |  | R | R |  | C |  |  | R |  | R |  |
| Add new boat |  |  | C |  |  |  |  | U | U | U | U |

Notes:

1. Pmnt=Payment
2. MNT=Maintenance job
3. CM=Crew member

### Use case diagram



## List of actors and descriptions

|  |  |
| --- | --- |
| **Actor** | Description |
| Unregistered user | * Register |
| Registered user | * Log in * Check into places * Add a place description * Add a comment |
| System | * Uploads data to server * Display places in X radius * Display other user location in X radius * Periodically update user location * Display GPS location * Display Time * Display comments * Display other user comments |

## Individual use case diagrams (TBC)

### Use case: Add a boat

|  |  |  |
| --- | --- | --- |
| Use Case id: | Add a boat | **CREATED BY** Tony Kaikeh |
| Actors: | Maintenance Staff |  |
| Brief Description: | This occurs when a new boat is added to the fleet | |
| trigger: | New boat purchased or manufactured | |
| includes: |  | |
| PreConditions: | A record of all boats available | |
| post conditions: | New boat added to database | |
| **BASIC FLOW** | | |
| 1. Staff initiates use case by adding boat details (boat name/ID) 2. system checks database for boat details 3. system confirms new boat details and requests confirmation 4. staff confirms boat is new 5. system adds boat to database 6. use case complete. | | |
| **ALTERNATIVE SCENARIOS** | | |
| 1. In step 2, system finds identical details for boat. System sends error message and ends use case. | | |
| **NOTES** | | |
|  | | |

### Use case: Check boat availability

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Use Case id: | Check Boat Availability | **CREATED BY** | Joe Vithayathil | |
| Actors: | Client, | | | |
| Brief Description: | The client checks the boat availability from the system. System then informs the user of the boats available on required date. | | | |
| trigger: | Booking enquiry | | | |
| includes: |  | | | |
| PreConditions: |  | | | |
| post conditions: | Booking complete and receipt sent. | | | |
| **BASIC FLOW** | | | | |
| 1. User enters booking requirements (no. of passengers, date and Boat class) 2. System checks booking schedule 3. System checks maintenance schedule 4. Systems informs user of available boats 5. End Use Case | | | | |
| **ALTERNATIVE SCENARIOS** | | | | |
| 1. In 1 date is outside holiday schedule. Error message and request new date. End UC. 2. In 1 required boat/s unavailable/on maintenance. Error message and request new date. End UC. 3. In 2 no boats available. End UC 4. In 3 boat requested booked for maintenance. Error message and request new date. End UC | | | | |
| **NOTES** | | | | |
| 1)Booking clerk is a specialised user  2) client is a generalised user | | | | |

### Use case: Make a payment

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case id: | Make a payment | **CREATED BY:** | Jolanta Rusecka |
| Actors: | CLIENT | | |
| Brief Description: | The Booking clerk receives the outstanding balance and then issues the appropriate document; a receipt if outstanding balance is paid. | | |
| trigger: | CLIENT | | |
| includes: | Generate Invoice and issue receipt | | |
| pre conditions: | Client must have a provisional Booking | | |
| post conditions: | Payment has been made and booking confirmation sent | | |
| **BASIC PATH** | | | |
| 1. Client initiates use case by entering booking reference number 2. System checks booking and confirms booking exists 3. Client requests making a payment 4. System calculates amount due and informs the Client 5. Client enters payment details 6. System confirms payment and sends the notification to the Client. 7. System updates booking status 8. The use case terminates | | | |
| **ALTERNATIVE PATH** | | | |
| 1. In step 2, Booking reference number wrong.  * 2a, Error message and use case ended.  1. In Step 2, Booking date past.  * 2a, Error message and use case closed.  1. In step 3, System declined request.  * 3a Use case terminates  1. In step 6, Payment not confirmed  * 6a Error message. * 6b System request payment details to be entered again. * 6c User inserts payment details. * 6d System confirms the payment * Use case closed. Use case continues at Step 7.  1. In step 6, Payment not confirmed.  * 6a Error message. * 6b System request payment details to be entered again. * 6c User inserts payment details. * 6d System rejects the payment * 6e Use case terminates | | | |
| **NOTES**  Client is a generalization of Booking Clerk | | | |

### Use case: Make a booking

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case id: | Make a booking | **CREATED BY** | Simon Board |
| Actors: | BOOKING CLERK | | |
| Brief Description: | Client calls to book a boat or makes a booking request at boat yard. The booker completes the booking form. | | |
| trigger: | CLIENT | | |
| includes: | Check boat Availability | | |
| pre conditions: | Booking/maintenance schedule available | | |
| post conditions: | Provisional Booking made | | |
| **BASIC FLOW** | | | |
| 1. The Booking Clerk initiates use case by entering the client’s details 2. the system records and checks client’s details 3. the system requests booking details 4. booking clerk enters booking details from form 5. the system initiates the check boat availability use case 6. the system notifies the clerk that boat/boats and weeks are available 7. system assigns a booking reference number 8. the booking clerk confirms booking details with the customer 9. the system flags booking as provisional and keeps it in a reserve state for 10 days 10. the system terminates the use case. | | | |
| **ALTERNATIVE SCENARIOS** | | | |
| 1. In step 2, client details already on system. System checks if booking already made for client 2. in step 4, booking dates out of holiday schedule. Error send and use case ends 3. in step 4, booking already made for client. Use case ends 4. in step 5, boats fully booked. Use case ends | | | |
| **NOTES** | | | |
| Each boat can be hired by the week up to a maximum of four weeks. Starts from 1400hr Saturday till 0930hr Saturday. | | | |

## Use case descriptions (TBC)

The head office which is in charge of scheduling holidays uses the **Schedule Holiday** use case at the beginning of the holiday season and if there are any changes to the pricing during the holiday schedule, they use the **Update Prices** use case.

The Booking clerk/clerk starts by using the **Make a Booking** use case to create a new booking using details obtained from a booking form. This use case includes the **Add Client** use case which records the client information if customer has never hired a boat and a **Check Boat Availability** use case to check the available boats and weeks for the client. Once the booking clerk adds the passenger details with **Add Crew Member** use case and selects an available boat for the desired week, the booking is marked as provisional. A deposit request is then sent to the client using the **Request Deposit** use case.

The client who then uses the **Make Payment** use case to pay for the hire. Once a deposit has been received the Accountant uses the **Generate Invoice** to send an invoice to. This then extends to the **Issue Receipt** use case, informing the client who confirms the amount paid.

If a client wishes to make any changes to the booking details, the **Amend Booking** use case is initiated, but this can become a **Cancel Booking** use case if the changes to the booking involve boat types or dates. This is then extended to a **Make a Booking** use case to create another booking.

If the **Cancel Booking** use case is started for a confirmed or fully paid booking the **Request Refund** use case is initiated to work out a refund for the client. This notifies the Head Office of the refund required. Having received approval from the Head Office, the Accountant uses the **Pay Refunds** use case to reimburse clients.

When all the bookings for the week have been confirmed/fully paid the office staff starts the **Create Booking Schedule** use case, to create and send the booking schedule to the maintenance yard. At the end of the week the Accountant initiates the **Produce Weekly Returns** use case, which sends all the weekly financial reports to the Head Office.

The Maintenance Department initiates the **Maintenance Schedule** use case to reserve requested boats and weeks for maintenance. This department also purchases or builds new boats and then uses the **Add a New Boat** use case to add a new boat to the fleet. Whereas a when a boat is damaged beyond repair, it is removed from the system using the **Write off a Boat** use case.

## E-R model (TBC)



**Descriptions of E-R model**



**Entity attribute table**

|  |  |
| --- | --- |
| **Entity** | **Attributes** |
| **Client** | Client\_no  Client\_type  Title  ClientName  ClientSurname  Postcode  AddressLine1  AddressLine2  Postcode  Town  ClientHomeTeNo(Compound Key)  ClientWorkTelNo |
| **CrewMem** | CrewNo  BookingRefNo\*  CrewMemTitle  CrewMemName  CrewMemSurname  CrewMemAge |
| **Booking** | BookingRefNo  BookingDate  Client\_no\*  NoOfAdults  Start\_date  Finish\_date  NoOfChildren  NoOfPets  Special req |
| **Boat** | BoatNo  BoatName  BoatDesc  ClassNo\*  Yard\_no\* |
| **Class** | ClassNo  ClassDesc |
| **Boats week** | WeekNo\*(Compound Key)  BoatNo\*(Compound Key)  BookingRefNo\*  MaintenanceNo\* |
| **Week** | WeekNo  WeekDate  NoOfWeeks |
| **Maintenance** | MaintenanceNo  OutOfSeasonMaintenanceStartDate  OutOfSeasonMaintenanceEndDate  DuringSeasonMaintenanceDate  EngineRunningHours  EngineerNameSurname  EffortIncurred  SpecialRepairDetails |
| **Payment** | Payment no  BookingRefNo \*  Deposit received(yes/no)  Deposit date  Payment date  CardType  CardNo  ExpiryDate  CardHolderName  CardHolderAddress  PostCode |
| **Price** | ClassNo (Compound Key)  WeekNo\* (Compound Key)  BoatPrice  VAT  TotalPrice |
| **BoatYard** | Yard\_no  Yard\_type  NoOfBoats |

## 

## Test plan and test cases

### Test plan

During analysis, the team uncovered a number of use cases and focussed on describing only four of our use cases from our model. During this phase we will be testing these use cases so as to be sure that they meet the business requirements. These use cases include

* check boat availability
* make a booking
* make a payment
* add a boat

To make sure that the new system will meet the requirements we will test our use cases in the following sequence:

* 1. add a boat
  2. check boat availability
  3. make a booking
  4. make a payment

1. **Use case: Add a boat**

Alt 1 starts at step 2, system finds identical details for boat. System sends error message and ends use case.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario | Originating flow | Alt flow  1 | Description | Test  case | Boat details correct | Are existing boat details identical to new boat? | Expected result |
| 1 | Basic  path |  | Normal scenario | 1 | Y | N | New boat added to system |
| 2 | Basic  path | Alt 1 | New Boat details identical to existing boat | 2 | Y | Y | Use case terminates. Error message sent. New boat not recorded |

1. **Use case: Make a booking**

Step 2 has 2 alternate paths where the system already has client details. If the client has made another booking then the use case ends (Alt 1b to step 10) if the client hasn’t made a booking (Alt 1a) go to step 3.

Step 4 has 2 alternate paths in step 4, Alt 2 indicates booking dates out of holiday schedule and use case ends. In step 4 Alt 3, booking already made for client then use case ends.

Step 5 has alternate path (Alt 4) where all boats are booked (unavailable). Use case ends.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario | Originating flow | Alt flow  1 | Description | Test  case | Are Client details existing | Is client on another booking | Requested Date is within season | Boats available | Expected result |
| 1 | Basic  path |  | Normal scenario | 1 | N | N | Y | Y | provisional booking confirmed. |
| 2 | Basic  path | Alt 1a | Clients details are already on system | 2 | Y | N | Y | Y | Client details entered. Booking completed |
| 3 | Basic  path | Alt 1b | Client details are on the system for another current booking | 3 | Y | Y | Y | Y | Use case terminates  Booking process is cancelled. |
| 4 | Basic  path | Alt 2 | Date requested is outside holiday season | 4 | N | N | N | N | Use case terminates.No booking is made. |
| 5 | Basic  path | Alt 3 | Client has already made a booking | 5 | N | Y | Y | Y | Use case terminates.No booking is made. |
| 6 | Basic  path | Alt 4 | All boats are unavailable | 6 | N | N | Y | N | Use case terminates.No booking is made. |

**3) Use case: Check boat availability**

**Test case description**

Step 1 actually has 2 Alternative scenarios where new date request is also included.

Step 2 Alt 2a requests new date when no boats are available on requested date. Back to Step 1.

Step 2 alt 2b no alternate date entered so End use case.

Step 3 alt 3 boats are under maintenance so end use case.

Step 3 alt 3b boats are under maintenance and alternate date is entered. Back to step 1

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario | Originating flow | Alt flow  1 | Description | Test  case | Boat available | Requested date in boat season | Boat is in maintenance on requested date | Alternate date entered | Expected result |
| 1 | Basic  path |  | Normal scenario | 1 | Y | Y | N | N | System displays boats available |
| 2 | Basic  path | Alt 1 | Date is outside season | 2 | N | N | Y | N | Use case terminates. Boat availability displayed unavailable |
| 3 | Basic  path | Alt 1a | Date outside season and alternate date is entered | 3 | N | N | Y | Y | Back to step 1 where system requests alternate date. |
| 4 | Basic  path | Alt 2a | No boats available on requested date. Alternate date entered | 4 | N | Y | N | Y | Back to step 1 where system requests alternate date. |
| 5 | Basic  path | Alt 2b | No boats available on requested date. No alternate date entered | 5 | N | Y | N | N | Use case terminates. Boat availability displayed unavailable |
| 6 | Basic  path | Alt 5 | Boat is in maintenance | 6 | N | Y | Y | N | Use case terminates. Boat availability displayed unavailable |
| 7 | Basic  path | Alt 6 | boats are under maintenance and alternate date is entered | 7 | N | Y | N | Y | System requests alternate date |

1. **Use case: Make a payment**

**Use case description**

Alt 1 starts at Step 2 is Booking reference is number wrong. Error message and use case ended.

Alt 2 starts at Step 2 is Booking has not been confirmed by the deposit payment so end use case.

Alt 3 starts at Step 3 system declines payment and ends use case.

Alt 4 starts at step 6 indicate Payment not confirmed and goes to step 3 where client re-enters payment details.

Alt 5 starts at step 6 where the payment details are entered and payment is rejected. This includes actions of payment being checked by system and client entering payment method. Use case ends.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Scenario | Originating flow | Alt flow  1 | Description | Test  case | Booking reference number correct | Deposit received | Payment details entered and confirmed | Payment accepted | Expected result |
| 1 | Basic  path |  | Normal scenario | 1 | Y | Y | Y | Y | Booking is confirmed.Use case complete |
| 2 | Basic  path | Alt 1 | Booking reference number is wrong | 2 | N | N | N | N | Use case terminates. Booking confirmation refused |
| 3 | Basic  path | Alt 2 | Booking deposit not received | 3 | Y | N | N | N | Use case terminates. Booking confirmation refused |
| 4 | Basic  path | Alt 3 | Payment declined | 4 | Y | Y | Y | N | Use case terminates. Booking confirmation refused |
| 5 | Basic  path | Alt 4 | Payment details not confirmed | 5 | Y | Y | Y | N | System asks for payment details to be re-entered |
| 6 | Basic  path | Alt 5 | Payment details not confirmed and Payment declined | 6 | Y | Y | N | N | Use case terminates. Booking confirmation refused |

## 6. Class diagram

## Appendices

Minutes of team meetings

Peer assessment matrix

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Print Name** | **Regular attendance at group meetings** | **Contribution of ideas for the task** | **Researching, analysing and preparing material for the task** | **Contribution to co-operative group processes** | **Supporting and encouraging group members** | **Practical contribution to end product** | **Individual marks to be allocated (X out of 24)** | **Signature** |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |
|  | **Grand Total** | | | | | |  |  |

**Agreement of contribution**

Grade each category on a five point scale from **0 = 'none' to 4 = 'excellent'**.

Name:

Signature:

Name:

Signature:

Name:

Signature:

Name:

Signature: