**Project Specifications for Mobile App Development**

# **Your Details**

|  |  |
| --- | --- |
| Name: | Joseph Lee |
| Class: | CC2004V |
| Reference Apps: | <https://outofdarts.com>  <https://toyrus.com.sg>  <https://nerf.hasbro.com/en-us> |

# 

# **Purpose**

Describe your app idea. What problem are you trying to solve with it? What objectives are you trying to accomplish? Tell us about the guiding principles that provide the basis for your app. Remember *KISS*

|  |
| --- |
| Users can buy Nerf blasters and darts. It will also provide facts and specs about the guns. The users can preview the gun with high quality images, view detailed specifications of the gun and detailed reviews from existing owners of the guns.  The app also aims to more accessible to parents allocating money for their children in events such as Christmas so that they can choose what they want by themselves. |

# 

# **Your audience**

Who are your potential users? Come up with 2 user stories for your application.

|  |
| --- |
| **Chang Shi, 16, Nerf Gun Enthusiast**   * As an enthusiast, he would want to find out more detailed specs for the app. Besides that, he would like to hear detailed reviews from other buyers including range, power etc. as well as delivery to multiple locations for his club and personal use   **Claudia, 32, Mum of 2 Children**   * She wants to get Christmas presents for her kids and wants to see the available catalogue of Nerf guns. She wants to let her children choose their gifts by setting up accounts for them and allocating a set amount of credit to each of her children. |

# Platforms & Technologies

Which devices would you like your app to work on?

|  |
| --- |
| The app should be able to run on mobile devices such as iPhone and iPad. Technologies used includes XCode and Swift for building app on iOS devices. |

# **Features and Functionality**

What do you want your app to do? What features do you want it to have? Tell us about all the things (login with social media and email, take and share photos and videos, make payments, determine location, analytics, scan codes, etc.) it should use? Please list them below

|  |
| --- |
| * Sell Nerf blasters and darts * Preview Nerf gun with high quality images * Give detailed reviews about the gun such as shooting speed, range, and durability * Recommend relevant blasters to the user * Buy darts for the blasters * Search for product based on name * Shopping cart for products user want to buy |

Translate the features into functional and non-functional specifications

|  |
| --- |
| A functional requirement describes **what a software system should do**, while non-functional requirements place **constraints on how the system** will do so.  Functional Requirements   * The app must be able login or register new users * The app must be able to complete purchases made by the user * The app must be able to reset account passwords by sending a reset email * The app must be able to add products into the user’s shopping cart * The app must be able to process the user payment by the provided app credits * The app must collect and store the user’s shipping address for delivery before payment   Non-functional Requirements   * The app should send the account confirmation within 2 hours of account creation * The app should send the user a receipt when they make a purchase on the app * The app should not crash when an error occurred, they should be reported to the developer instead for bug fixing * The app should be able to display high quality images of the products within 5 seconds * The app size should not exceed 50 megabytes * The app should have a search function that can search for products in the shop * The app font size should be appropriate for different screen sizes. It should not be too large on devices with small screen size |

# **Admin Features**

Describe how you would like to manage the application as an administrator. What functions do you want to be available on the admin panel? Please list them below

|  |
| --- |
| * Export product sales as pdf/excel formats * Manage user information * Manually reset user passwords * View user purchases for packing and be able to change delivery status * CRUD functions for products in shop * Create sales on certain products * Edit home page carousel banners images |

# **Designs/Wireframes/Mockups**

Insert your mockup/ wireframe here

|  |
| --- |
| **Initial Wireframes**    **Mockup**  Graphical user interface, application  Description automatically generated |

# **Timeline & Budget**

What date have you planned the release for? Have you already done some budgeting? Please tell us about any key dates and numbers we should know about

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| The project should take around 6 months to design and develop. Estimated budget for the product will be around $15,000 - $25,000   |  |  |  |  | | --- | --- | --- | --- | | Development & Maintenance Costs | | | | | Name | **Type** | **Amount** | **Notes** | | Initial Development | One-time | $15,000 - $25,000 |  | | Server Costs | Monthly | $200 - $350 | For 100-200 concurrent users | | Continuous Maintenance & Development | Monthly | $2,000 | Bug fixing and additional support | |

# **Acceptance Criteria**

Please tell us about the essential requirements to the app performance that you absolutely want to see in the final release. If there are some test cases or scenarios where the app acceptance criteria can be tested and measured, please let us know about them

|  |
| --- |
| * The app must be able complete purchases made to the user * The users must be able to sign up for an account to shop. * The user must be able to add items into their cart for purchase. * The user must be able to reset their account password should they forget it. * The user must be able to search and filter products. * The user must be able to add, edit and delete their delivery addresses * The user must be able to check for delivery estimates based on weather conditions |

# **Extra Information**

If you’d like to add anything else, please complete the field below

|  |
| --- |
| Product images taken from Toy’r’US  <https://toyrus.com.sg> |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Test Scenario | Test Steps | Expected Results | Actual Result | Pass/Fail |
| TU1 | Successfully sign-up user | 1. Open the app 2. Tap the register button 3. Enter the user details in the form (Username: user1 | Email: [user1@email.com](mailto:user1@email.com) |Password: user) 4. Press the register button | The user should successfully register into the store database and an alert will appear | Graphical user interface, text, application  Description automatically generated  The account has been created and the alert appears | Pass |
| TU2 | Login validation works (wrong password, correct user | 1. In the log in screen fill in the form with the following details  * Username: user1 * Password: user1 | An alert should appear with a message that the password or username is incorrect, and the user should not be logged in. | Graphical user interface, application  Description automatically generated  The app showed an alert with an error that the username or password is incorrect | Pass |
| TU3 | Successfully log in into the app | 1. On the login screen, log in with the following user details  * Username: user1 * Password: user | The app should log in the user and the home screen should appear | A screenshot of a phone  Description automatically generated with medium confidence  The app logs the user in, and the home view appears | Pass |
| TU4 | Get a list of products | * On the home screen, Tap on the ‘Nerf Blasters’ button | The category screen should appear with a product list related to the category | Graphical user interface, application  Description automatically generated  The category view appeared and the product list for that specific category is shown | Pass |
| TU5 | Get a product detail | * Tap on the ‘Nerf Blasters’ button * On the category page, tap on any product to bring up the product details | The product details page should appear with the product details of the correct product that the user has requested for. | Graphical user interface, website  Description automatically generated  The product detail is displayed with the correct product image, title and description | Pass |
| TU6 | Search for a product and view its product detail | * In the Home screen, tap the search bar * Input ‘x-shot’ as the search term * Tap on the first product listed in the results | The search results should be relevant to the results and the product detail should appear with the correct information requested by the user. | Text  Description automatically generated with medium confidenceGraphical user interface, text  Description automatically generated  The search results is shown with relevent products and when the user clicks on the first product in the list, the product detail view appears with the correct product image, name and description | Pass |
| TU7 | Insert a product into the cart (Single Item) | * Select ‘Nerf Blasters’ on the home screen * Select the first product in the Nerf Blasters category * In the product detail screen, tap on ‘Add to cart’ button | An alert should appear telling the user that the item has been added to the cart | Graphical user interface, website  Description automatically generated  The product has been added to the cart and the alert appears | Pass |
| TU8 | View the cart | * In the home screen, press the cart icon on the right-hand side of the screen | The cart screen should appear.  According to the previous test case (TU7), the user should have one item in the cart | Graphical user interface, application  Description automatically generated  The cart view opens with 1 item inside due to the previous test | Pass |
| TU9 | Insert a product into the cart (Multiple Items) | * Select ‘Nerf Blasters’ on the home screen * Select the second product in the Nerf Blasters category * In the product detail screen, type in ‘3’ in the quantity text field * Tap on ‘Add to cart’ button * Go back to the home screen and open the cart | An alert should appear telling the user that the item has been added to the cart  When the cart is opened, the newly added product should be displayed with the correct quantity and price | Graphical user interface, text  Description automatically generatedGraphical user interface, text, application  Description automatically generatedGraphical user interface, application  Description automatically generated  The selected product has been added to the cart. When the user opens the cart, the product is inside with the correct quantity and price | Pass |
| TU10 | Insert a product into the cart (Same item twice) | * On the home screen, tap on the ‘Nerf Blasters’ button * Select the third product in the Nerf Blasters category * Tap on the ‘Add to Cart’ button * Dismiss the alert * Tap on the ‘Add to Cart’ button again * Go back to the home screen and open the cart | An alert should appear telling the user that the item has been added to the cart  When the cart is opened, the newly added product should be displayed with the correct quantity and price  The item quantity should be 2 and it will not be displayed as two separate items | Graphical user interface, website  Description automatically generatedGraphical user interface, application  Description automatically generated  The product has been added to the cart twice however since it’s the same product, it will just increment the product quantity and display the correct price. | Pass |
| TU11 | Delete a specific item in the cart | * In the home screen, press the cart icon on the right-hand side of the screen * Swipe left on the first item in the cart | The item should get deleted from the cart and the total price should get re-calculated | Graphical user interface, application  Description automatically generatedGraphical user interface, application  Description automatically generated  The item has been deleted from the cart and it disappears from the cart list. The total price has also been re-calculated to reflect the new total price. | Pass |
| TU12 | Edit an item quantity | * In the home screen, press the cart icon on the right-hand side of the screen * Swipe left on the first item in the cart * When the alert appears, select the text field inside and edit the quantity to 5 * Tap on the ‘Edit’ button to save the changes | The quantity should get edited to the correct amount.  The price should get re-calculated to reflect the changes | Graphical user interface, application  Description automatically generatedA screenshot of a phone  Description automatically generated with medium confidenceGraphical user interface, application  Description automatically generated  The product quantity is changed and the total price is re-calculated to reflect the changes. | Pass |
| TU13 | Edit an item quantity to 0 (Remove item via edit quantity) | * In the home screen, press the cart icon on the right-hand side of the screen * Swipe left on the first item in the cart * When the alert appears, select the text field inside and edit the quantity to 0 * Tap on the ‘Edit’ button to save the changes | The item should be removed from the cart and the item should disappear from the cart list  The price should get re-calculated to reflect the changes | A screenshot of a phone  Description automatically generated with medium confidenceGraphical user interface, application  Description automatically generated  The product is deleted from the cart and is also removed from the list. The total price is re-calculated to reflect the changes. | Pass |
| TU14 | Delete entire cart | * In the home screen, press the cart icon on the right-hand side of the screen * Press the delete cart icon on the right side of the navigation bar * Press ‘Delete’ when the alert appears to confirm the deletion | The cart should get deleted, all items in the cart screen should disappear and the total price should get reset to 0 | A picture containing graphical user interface  Description automatically generated  The entire is successfully deleted and the list disappears. The total price resets to 0 as well. | Pass |
| TU15 | Viewing checkout page | * In the home screen, press the cart icon on the right-hand side of the screen * Press the checkout button in the cart page (Assuming from previous test cases, there will be items in the cart) | The checkout page should appear, and the following information should be shown   * The cart payable amount * The user credit amount along with the final amount expected after deduction   All information should be accurate | Graphical user interface, application  Description automatically generated  The checkout view appears. The total price, avaliable credits and the final amount after deduction is shown correctly. | Pass |
| TU17 | Checkout discount shake (Received >0% discount) | * In the checkout screen, shake the mobile device to trigger the discount | The app should apply a discount to the checkout total price and deduct the total cost based on the percentage amount.  An alert should also appear telling the user that a discount has been applied to the checkout | Graphical user interface, application  Description automatically generated  The discount has been applied to the checkout and the alert appears stating that the discount with the amount has been applied |  |
| TU18 | Checkout discount shake (Received 0% discount) | * In the checkout screen, shake the mobile device to trigger the discount | No discount should be applied to the checkout total price. An alert will appear telling the user that no discount has been added to the checkout | Graphical user interface, application  Description automatically generated  An alert appears stating that no discount has been applied |  |
| TU19 | Top-up credits (2000 credits) | * In the checkout screen, press the ‘Top-up Credits’ button below the credit balance * In the top up credits page, select the text field and type in ‘2000’ * Press the ‘Add Credits’ button to confirm the top up | The credit wallet should get topped up with the requested amount and the app will return to the checkout page  The checkout page will reflect the credit wallet value with a new final balance after payment  In this case 200 + 2000 = 2200 Credits | Graphical user interface, application  Description automatically generatedGraphical user interface, application  Description automatically generated  The credits has been successfully been added to the account and when the app returns to the checkout view, the avaliable credits and final balance after payment updates to reflect the new amount. | Pass |
| TU20 | Top-up credits validation (100000 credits) | * In the checkout screen, press the ‘Top-up Credits’ button below the credit balance * In the top up credits page, select the text field and type in ‘100000’ * Press the ‘Add Credits’ button to confirm the top up | An alert will appear with an error telling the user that the amount exceeds the maximum allowed amount. | Graphical user interface, application  Description automatically generated  An alert with an error message appears telling that the amount has exceeded the maximum top-up amount | Pass |
| TU21 | View Delivery Addresses (No addresses) | * In the checkout screen, press the ‘Select address’ button to bring up the address list screen | The address list screen should appear and only one row will appear telling the user that no delivery address has been added yet. | A picture containing text  Description automatically generated  The address list appears with only a prompt stating that there are no delivery address added yet. | Pass |
| TU22 | Add an Address | * In the address screen, select the plus button located on the right side of the navigation bar * In the form, fill in the address details accordingly * Name: ITE College Central * Address Location: 1 Ang Mo Kio Dr, Singapore 567720 * After filling in the details, press the ‘Add Address’ button to save the address * Click ‘okay’ on the alert | The delivery address gets added into the app database and an alert appears to tell the user that the address has been added  After the alert has been dismissed, the app should return to the address list screen with an updated list of delivery addresses | Graphical user interface, application  Description automatically generatedA picture containing text  Description automatically generated  The address has been successfully added into the app and an alert appears stating it.  After dismissing the alert, the app returns to the address list where it updates with the new address included in the list | Pass |
| TU23 | Add address form validation | * In the address screen, select the plus button located on the right side of the navigation bar * In the form, fill in the address details accordingly * Name: ITE College Central * Address Location: [Leave this blank] * After filling in the details, press the ‘Add Address’ button to save the address | An error alert should appear telling the user that not all fields have been filled up | Graphical user interface, application  Description automatically generated  An alert with an error appears to tell that not all fields are not filled in. | Pass |
| TU24 | Edit Address (Swipe) | * In the address screen, swipe right to edit the address * In the edit address screen, select the ‘Address location’ text field and change the existing location to ‘2 Ang Mo Kio Dr, Singapore 567720’ * Click the save icon on the right side of the navigation bar to save the changes | An alert should appear telling the user that the address has been edited.  After the alert is dismissed, the app should go back to the delivery addresses list and should be updated with the edited information | A picture containing graphical user interface  Description automatically generatedA picture containing text  Description automatically generated  The address has been updated and the address list updates to reflect the new changes. | Pass |
| TU25 | Select an address and preview delivery timings | * In the address list screen, select the first address in the list (Should be ‘ITE College Central’) | The app should return to the checkout screen with the address set.  Along with that, the delivery timings should start getting data online to predict delivery timings. Once the data is received, it will show ‘Same day’ if it is not raining at the delivery location or ‘Next day’ if it is raining at the delivery location | Graphical user interface, application  Description automatically generated  The app returned to the checkout view with the address set in the ‘Deliver To’ section | Pass |
| TU26 | Complete purchase in checkout | * In the checkout screen, make sure to have enough credits and set a delivery address * Click the ‘Complete payment’ button to complete the purchase | A success page should appear, and the credit wallet balance deducted from the purchase | Graphical user interface  Description automatically generated with low confidenceGraphical user interface, application  Description automatically generated  The checkout was successful and the correct amount of credits has been deducted from the account. | Pass |
| TU27 | View profile tab | * In the home screen, select the ‘Profile’ tab at the bottom centre | The profile view should be displayed, and the correct user information should be displayed  In this case (according to previous test cases):   * Username: User1 * Email: [user@email.com](mailto:user@email.com) | Graphical user interface, application  Description automatically generated  The account view displays with the correct user information such as email and username | Pass |
| TU28 | Edit profile (email) | * In the profile screen, select ‘Change E-mail Address’ from the list of account options * Fill up the form with the following details * Email Address: [user1@email.com](mailto:user1@email.com) * Current password: user * Press the ‘Save’ button to confirm the change | The app should overwrite the old email with the new email that has been inputted and an alert should appear telling the user that the email address has been successfully edited  When the alert is dismissed, it should return to the account screen where the email address would update. | Graphical user interface  Description automatically generatedGraphical user interface, application  Description automatically generated  The account e-mail address is successfully changed and an alert appears stating it.  When the alert is dismissed, the app returns to the account view with the e-mail address updated to the new one. | Pass |
| TU29 | Edit profile validation (wrong existing password) | * In the profile screen, select ‘Change E-mail Address’ from the list of account options * Fill up the form with the following details * Email Address: [user1@email.com](mailto:user1@email.com) * Current password: aaaaa * Press the ‘Save’ button to confirm the change | An alert with an error should appear telling the user that the existing password does not match the existing one | Graphical user interface, application, website  Description automatically generated  An alert appears with an error telling that the existing password is incorrect and therefore unable to save. | Pass |
| TU30 | Edit profile validation  (Missing fields) | * In the profile screen, select ‘Change E-mail Address’ from the list of account options * Fill up the form with the following details * Email Address: [[Leave this blank]](mailto:user1@email.com) * Current password: aaaaa * Press the ‘Save’ button to confirm the change | An alert with an error should appear telling the user that not fields has been filled up | Graphical user interface, website  Description automatically generated  An alert appears with an error telling that not all field in the form are not filled up. | Pass |
| TU31 | Logout Account | * On the home screen, press the ‘X’ button on the top-left side of the navigation bar * When the logout alert appears, select the logout option to logout from the account | The app should logout the user from the app and clear user data. The app will then send the user back to the login screen.  When the user closes and open the app again, it should not automatically log in the user and the login screen should appear instead. | Graphical user interface, application  Description automatically generatedGraphical user interface, application  Description automatically generated  The user is logged out from the app and the login screen appears |  |
| TU32 | Delete Account | * In the profile screen, select ‘Delete Account’ from the list of account options * Press the ‘Delete Account’ button to confirm delete * Press ‘Delete’ in the alert to confirm once more | Account should get delete and the user will get send back to the login screen  When the user tries to login into the deleted account, the app should not allow login | Graphical user interface, application  Description automatically generatedGraphical user interface, application  Description automatically generatedGraphical user interface, application  Description automatically generated  The account get deleted and the app returns the user to the login view.  When the user tries to login with the old account again, it did not allow the user to login as the user does not exist. | Pass |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Screenshots | | | | | |
| Graphical user interface, application  Description automatically generated  Login | Graphical user interface, application, website  Description automatically generated  Register | A screenshot of a phone  Description automatically generated with medium confidence  Home | Graphical user interface, application  Description automatically generated  Account Options | Text  Description automatically generated  App Options | Graphical user interface, text, application  Description automatically generated  Categories |
| Graphical user interface, text, application  Description automatically generated  Category | Graphical user interface, application  Description automatically generated  Product Details | Graphical user interface, application  Description automatically generated  Cart | Graphical user interface, application  Description automatically generated  Checkout | Graphical user interface  Description automatically generated with low confidence  Address List | Graphical user interface, text  Description automatically generated with medium confidence  Add Address |
| Graphical user interface, application  Description automatically generated  Add credits | Graphical user interface  Description automatically generated with medium confidence  Checkout successful | Graphical user interface, application, website  Description automatically generated  Edit Email/Password | Graphical user interface, application  Description automatically generated  Edit product quantity (swipe) | Graphical user interface, text  Description automatically generated  Edit address (swipe) | Graphical user interface, application  Description automatically generated  Sort addresses |
| Delete Product (Swipe) |  |  |  |  |  |