TABLE OF CONTENTS

1. Introduction

1.1. Document overview

This document is the software test plan of the Custom Computer Express Mobile App software development project. It contains the list of tests, which are executed during the phases of Custom Computer Express Mobile App integration and verification:

- Software Integration tests
- Software Verification tests

1.2. Group Members

Christopher Bickelhaupt
Justin Sosa
Evan Wan
Beverly Wertz

2. Test environment

2.1. *Hardware test Platform*

The hardware available to test on will be a mixture of emulated and physical devices in the form of tablets and phones:

- The Hardware emulation tests will be run on a simulated AVD Android phone with the following specifications:
  - Screen with WVGA800 format
  - Simulated ARM Processor
  - 256MB memory
  - 10GB storage space
  - Wi-Fi
  - Android 2.2
- Available android phones
  - Galaxy S & Galaxy S II
  - Samsung
  - Android 2.3.5
  - IGH987
  - 256MB memory
  - 3.5” AMOled screen
  - 16GB Storage

2.2. Software test tools

The Software Test Tools that will be used are:

- Eclipse IDE Java SE Version
- Android SDK with AVD manager
- Shared Dropbox file repository
- Eclipse JUnit testing

2.3. *Test Data and documentation*

The following test data will be used:

- Test responses to represent gaming users, light internet users, educational users
- Test configurations will be generated that represents invalid and valid configurations
- Test address will be used to represent valid and invalid addresses/name input

The results will be produced in a results output text file on the local machine running the tests.

2.4. *Installation, set-up, and maintenance*

Installation of software on local team member machines will be required along with updates to ensure cross-system compatibility.

2.5. Personnel

The individuals performing the tests will be members of the Custom Computer Express group developing the project.

3. Tests identification

3.1. *Testing phases*

This test plan defines all tests to verify all requirements of Custom Computer Express Android application software in the following successive testing phases :

- Unit tests
- Integration tests
- End-user or customer tests.

3.2. *Test categories*

Tests are distributed in categories, depending on the tests performed:

- Risk analysis mitigation tests
- Main functions
- Response time
- User case studies

3.3. *Test progression*

The tests' progression depends on the testing phase:

- Unit tests:
  - The testing tool automatically sets the test progression. There is no dependency between unit tests.
- Integration tests: tests are executed according to the following rationale:
  - Test each individual Android activity
  - Integrate all activities with the Android application
- End-user tests:
  - Test progression is defined according operational scenarios.

3.4. *Test coverage*

Describe tests coverage rationale. Example

Tests coverage depends on the testing phase:

- Integration tests cover all interfaces requirements of CCEsoftware
- Alpha tests cover all requirements defined in the SRS, excepted ….
- Beta tests cover all requirements defined in the SRS

3.5. *Data recording, post-processing, and analysis*

- Raw test data will be recorded in a test data file. All tests will be labeled by their test identification ID.
- The test data file will be shared will all team members.
- Test data will be checked through automation.

3.6. *Test identification and content*

Each test is unique and contains:

- A unique identifier
- The tests category
- A textual description of test objective
- The traceability of the SRS requirement(s)
- Assumptions and constraints, if any
- Safety, security and privacy concerns, if any.

The identifier has the following structure:

- T-Description

4. Planned Tests

For each phase, a list of tests is defined with an order of execution if necessary.

4.1. *Tests Phase Alpha*

4.1.1. Tests coverage

The tests of phase alpha cover the following range:

- Interfaces and critical requirements
- Functional Requirements

4.1.2. Planned tests

Planned tests of phase alpha are listed in the table below. They are executed in the same order.

Identifier	Description
T-Interface Navigation	Verify that each screen of the application navigates to the correct destination.
T-Cart and total	Verify that the cart is filled with the correct parts and update accordingly. Verify that the running total is updated according to the cart.
T-Error Checking	Verify that all required buttons and forms are filled out correctly

4.2. *Tests Phase Beta*

4.2.1. Tests coverage

The tests of phase beta cover the following range:

- All requirements of the SRS
- A functional domain
- All requirements

4.2.2. Planned tests

Planned tests of phase beta are listed in the table below. They are executed in the same order.

Identifier	Description

T-Speed of navigation	Verify that the interface transitions and user submissions are completed within a specific timeframe.
T-Payment Verification	Verify that the payment function works correctly by validating user payment info
T-Image display	Verify that all the required images (logos, computer parts, etc.) display correctly.
T-Confirmation Number	Verify that the confirmation given to users after a completed purchase is correctly stored by the software for later access

5 Completed Prototype Testing

5.1 Interface Testing

Main Menu Testing:

- Checked all menu buttons to make sure they are clickable and display the correct animations
- Checked all menu buttons to make sure they lead to their corresponding interface
    - Pre-Built Computers
    - Help Me Buy a Computer
    - Configure and Purchase a Computer
    - Check Order Status
    - Exit Application
- All content and text are displayed correctly on the Main Menu Screen
- All user info is pulled from the database upon entering the Main Menu Screen

Pre-Built Computer Testing:

- Checked to make sure all the buttons are working properly
- All computers display their correct parts, computers, and images
- Checked to make sure that when selecting a pre-built computer that it is added to the cart
- Checked to make sure when selecting a computer to customize that all the parts are pre-set in the customization menu.

Help Me Buy A Computer Testing:

- Checked to make sure all buttons are working properly
- All the questions are displaying the right text to the user
- Checked to make sure the rating bar is working properly and the ratings are stored appropriately
- Checked to make sure there is error checking on the questions
- Checked to make sure the correct parts for their corresponding questions are being added to the cart

Configure and Purchase a Computer Testing:

- Checked to make sure all the buttons are working properly
- All the radio buttons are working properly and are highlighting.
- Checked to make sure all the view-part buttons are displaying the right images and text
- Checked to make sure the help buttons are displaying the right content
- Checked to make sure there is appropriate error checking on each page
- Checked to make sure the part that is selected is put in the cart correctly

Check Order Status:

- Checked to make sure all the buttons are working properly
- Checked to make sure all the queries are being pulled properly
- Checked to make sure user data is being added to the database

Cart Summary Screen

- Checked to make sure all the appropriate prices are listed for their corresponding parts
- Checked to make sure the total price is calculated properly

5.2 Database Testing

Queries:

- Checked to make sure all the queries are modifying, deleting, and inserting properly
- Checked to make sure the appropriate data is being stored in each table

Reports:

- Checked to make sure the report on the Cart Summary Screen is working properly and pulling appropriate data

Maintaining:

- Checked to make sure none of the data is getting erased or over written
- Checked to make sure that data in shared preferences is being passed from interface to interface

5.3 Usability Testing

To achieve usability testing, we asked a couple of people to test our application out in front of us. While they were testing the app, they were required to say out-loud everything they were thinking while using it. Based on some of the results we made significant design changes. Here is the list of problems that most users were having:

- Not enough help on the customization screen.
- Options on customization screen were difficult to choose from for a novice user
- Button layout on the screen was unorganized and buttons were too big

In order to fix these 3 problems were implemented the following solutions:

- Added help features on the customization screen that gave detailed information about each part
- Added pictures of each part on the customization screen so that users could know what they were buying
- Added a style to the button layout that became uniform for the whole application

6 Requirements traceability

Identifier	Description	SRS Requirement
T-Speed of Navigation	Verify that the interface transitions and user submissions are completed within a specific timeframe.	SRS-NF-4
T-Payment Verification	Verify that the payment function works correctly by validating user payment info	SRS-NF-2 SRS-FR-6 SRS-FR-7
T-Image display	Verify that all the required images (logos, computer parts, etc.) display correctly.	SRS-NF-4
T-Confirmation Number	Verify that the confirmation given to users after a completed purchase is correctly stored by the software for later access	SRS-FR-3

7. Risk Assessment and Management Plan

Risk	Likelihood	Impact	Risk Management Approach/Mitigating Actions	Early Warning Signs
Skills				
Lack of in-house skills at initial stages	High	High	Team members will need to become knowledgeable in the Android platform.	Team member are having trouble working through the Android API
Loss of key staff	Low	Med	All code should be commented and documented	
Not meeting the needs of staff with different levels of skill	Med	Med	Team members will need to help mentor each other and take the initiative to catch up on skill that they lack	
Management				
Failure to get all parties to share same understanding of purpose	Med	High	All parties should have a clear understanding and a shared purpose before implementation of the application. All misunderstandings should be worked out in the design phase.	Differing views on forward plan.
Failure to get team members to meet	Low	High	All team members need to know the impact they have on coming to meetings and know the consequences of excessive absence.	Team members start not showing to meetings and give excessive excuses for meeting attendance.
Failure to complete assignments on schedule	Low	High	Workload of members should be appropriate and work should be assigned and completed in a timely manner.	Team members have trouble completing their assigned work in the designated time.
Infrastructure				
IT infrastructure cannot cope with requirements	Low	High	Have multiple backup computers available to assist in recovery should one fail.	Materials not working, speed of equipment response
Cellular network becomes unstable/unusable	Low	High	Ensure vital data is kept secure until the network comes back online	Speed of network response in app functions
Delivery				
Staff not available at meeting times	High	High	Flexible delivery and sessions on different days and at different times	Timetables. On-line meetings should be an option.