**hCS673F13 Software Engineering**

**Group Project 3 - Edu kid**

**Project Proposal and Planning**

|  |  |  |  |
| --- | --- | --- | --- |
| Team Member | Role(s) | Signature | Date |
| Kevin Graue | Leader, Environment and Integration Leader | *Kevin Graue* | 9/21/2013 |
| Sweekriti Satpathy | Co-Design Leader | *Sweekriti Satpathy* | 9/22/2013 |
| Sandeep Nukala | Configuration leader | *Sandeep Nukala* | 9/23/2013 |
| Isha Rana | Quality Assurance Leader | *Isha Rana* | 9/25/2013 |
| Rakshit Sachdev | Co-Design Leader | *Rakshit Sachdev* | 9/25/2013 |
| Jasjot Singh | Implementation Leader | *Jasjot Singh* | 9/25/2013 |
| Peter Trevino | Backup team Leader, Requirement leader | *Peter Trevino* | 9/25/2013 |
| Chitra Premkumar | Quality Assurance Leader | *Chitra Premkumar* | 9/25/2013 |

**Revision history**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Date** | **Change** |
| 1.0 | Kevin Graue | 9/21/2013 | Initial creation |
| 1.1 | Chitra Premkumar | 10/09/2013 | Quality Assurance Plan,Detailed overview |
| 1.2 | Isha Rana | 10/09/2013 | Quality Assurance Plan,Overview,Detailed description |
| 1.3 | sandeep | 10/18/2013 | Configuration Plan Management |
| 1.3 | Chitra Premkumar | 12/07/2013 | Risk management Plan |
| 1.4 | Isha Rana | 11/23/2013 | Final completion and editing |

[Overview](#h.87t9hln2vjz0)

[Related Work](#h.mps353x5ezyl)

[Detailed Description](#h.fg3z0hpd4q9v)

[Management Plan](#h.ds8oyr75pnh1)

[Process Model](#h.27177f40uci)

[Risk Management](#h.a4oqwntk3mw)

[Monitoring and Controlling Mechanism](#h.ywdoc2clc9yt)

[Schedule and deadline](#h.tadq5mb0pici)

[Quality Assurance Plan](#h.72e1f4uawy2r)

[Metrics](#h.b2haznn3yyz2)

[Standard](#h.vc72k6dweldv)

[Inspection/Review Process](#h.f1c69ifi68h7)

[Testing](#h.r5d5mhtlf0kq)

[Defect Management](#h.54a4wuncjg1c)

[Process improvement process](#h.jhct37ebxxpn)

[Configuration Management Plan](#h.hw41vg4ykxen)

[Configuration items and tools](#h.bwlb4d4vdox2)

[code commit guidelines](#h.yyauft6zr9hw)

[References](#h.8mva2050iy7t)

[Glossary](#h.ty3i2nqffhtc)

# Overview

EDUkid is an Android App, to be used primarily on the tablet. Our mission is: "Delight children through learning." The motivation of this app to create an application which not only allows children to learn in a fun and interactive way, but to also allow parents to be involved in the project. We want to allow parents to be able to customize the content as per their requirement and needs of their child.

Our main purpose is to help making learning more enjoyable for the child and also integrate the current world advancement of technology of tablets into children’s education field.

We aim to make this app available for children in the age group of 5 and above and for their parents.

# Related Work

In the current market scene, there are various children’s application available on the Android PlayStore and even on other platforms like Apple’s iOS. But what makes our app stand out from the rest apart from the others is:

* Learning alphabets, numbers, colors and shapes using real-life objects and a customizable feature added for parents
* The game does not overload children and parents with too many audio and visual stimuli. Coping with stress and over-stimulation is therefore one skill that your kids won't be practicing during this game. Its clear focus makes the game a delight for children and a winning choice for parents.
* Educational games for young children is all we do. Our mission is: "Delight children through learning."
* Our games are laser focused. For example, the numbers game doesn't teach letters, and the letters game doesn't teach math. We keep the games simple but magically inviting and nourishing.
* We strive to give children just the right balance between education and fun. So our games don't feature fun at the expense of education – or education at the expense of fun. We also know games that are too complicated don’t involve and delight kids.
* We have included features like customizable menus, play-timer which makes this app a complete delight for parents allowing them to control the usage time of their children

# Detailed Description

EDUkid is an educational application aimed to make learning fun and interactive for a child.

The app features include:

* Category wise arrangement of available games in the default categories Alphabets, Numbers, Colors and Shapes.

Small Settings Button with a “Question Based Passcode” which entails the parent to solve a high level question to enter the settings menu.

* Bigger Buttons with Simplistic UI enabling the child complete use of the app
* Timer Mode where a parent can set the time for which a child can play the games
* Fun and Interactive Games to test the child
* Capturing child’s image through the Camera button provided on home screen
* Timer provided to ensure that child does not spend too much time in playing the game
* Customization is possible upto a great extend:

->Parent can add images of their own choice by directly uploading

them or capturing an image using the button provided to do so.

->Audio can be recorded by the parent

The Application makes complete use of the Android Tablet and features available from the operating system:

* Gesture controlled menus and games
* Use of the Voice and Audio features
* Text to Speech for creation of Child Profile

# Management Plan

## Process Model

We will be using an iterative process model consisting of 4 iterations throughout the semester.

The team member roles are as follows:

**Kevin** – Project Leader and Environment and Integration Leader

**Peter** – Backup Project Leader and Requirement Leader

**Sandeep** – Configuration Leader

**Sweekriti** – Design Co-leader

**Rakshit** – Design Co-leader

**Jasjot** – Implementation Leader

**Isha** – QA Co-leader

**Chitra** – QA Co-leader

## Objectives and Priorities

Our main objective is to have a fully functional Android App at the end of the 4th iteration. The app will assist a child in learning basic ideas such as the alphabet, numbers, shapes, colors, etc. These main features are essential to the app and must be included given the fixed cost (8 team members working a handful of hours per week) and the fixed deadline (current semester).

We have a list of features that are not critical to the app release that we will include if time permits. These include an NFC feature for scanning items to add functionality to the app, and accomplishments/leaderboard view to show learning progress, and a game mode to test the child on their current knowledge.

## Risk Management

The risks for this project include, but are not limited to:

* Completion of planned functionality
* Comprehension of new tools and development kits that may not be familiar with the entire team
* Ability for a large team of 8 to be productive and work together
* Finding common timing for group meeting.
* Resolution of images in different devices.
* Database management.
* Lack of camera feature in user’s device.

...

## Monitoring and Controlling Mechanism

Weekly status will take place every Thursday night in GCB 209 following class. The meetings will take some ideas from Agile Scrum. At the end of each meeting each member will speak in a “standup” fashion to discuss what they worked on since the last time we met, what they are working on (or plan to work on), and any issues they might be having that can be resolved with the help of another team member.

## Schedule and deadlines

The team consists of 8 team members each working an average of 5 hours per week. Starting with Iteration 2, we will have 9 weeks to complete the work. This comes to a total of approximately 360 hours.

# Quality Assurance Plan **Metrics**

**i. Product metrics**

* Size
* Performance
* Reliability

**ii. Process metrics**

* Defect density
* Defect fixing
* Requirement stability
* Efficiency

## Coding Standard

* Standard Java coding practice
  + Camel case for variables (example: thisIsAnExampleVarible)
  + Class names always start capitalized (example: ExampleClass.java)
  + Root package: bu.edu.cs673.edukid
  + Use standard Eclipse code formatter before code commits

## Inspection/Review Process

## Self Inspection

* + individual tracking of our assigned task.

## Team Inspection

## As a part of team, a member has to do the tracking of the project after each iteration

## 

## Testing

* **Unit testing**

We have iterative testing with a QA designed test script.We stored the testing script as a google spreadsheet.If given access, we did User Acceptance Testing with a child after we successfully completed the QA test script.

This required the execution of the software or parts of the software. The software can be executed in the target system, an emulator, simulator or any other suitable test environment. Within the range of dynamic tests the state of the art distinguishes between structural and functional tests. The structural dynamic tests are also called "module tests" or "unit tests". A module test is performed with the knowledge of the module internals in mind. I.e. especially the branches and paths in functions and modules.

Before each commit checking of all features on testing spreadsheet is done to ensure nothing is broken ,followed by add any additional features to testing spreadsheet ,committing to current testing branch and notifying testing partner that code has been committed to testing branch .Testing partner verifies all existing functionality as well as added functionality followed by notifying branch maintainer that all features have been tested and double checked.Finally testing branch is merged to main branch.

We followed an amalgamation of TDD and Integration testing

* **Testing Branch**

All new features are tested in the testing branch(dev). The branch maintainer checks the code and merges the testing branch to master branch.

* **JUnit Testing for Settings Page Password**

Android Test Cases are based on Junit.Test package is created along with the android application

Two methods created for testing code for settings page password.

***testcorrectMathProblem***test method we compare answers with *assertEquals*

***testfalseMathProblem***test method we compare the answer and *Max value* of integer with *assertFalse*

* **Database Testing**

Database is build using SQLiteOpenHelper.

Database testing is also done using JUnit Testing.

## Defect Management

* + 1. We will be using a defect management table
    2. Priority will be a composite score based on impact and urgency
    3. Urgency will be scored based on severity of bug, will be scored between 0 and 3
    4. Impact will be scored based on perceived impact to user. Will it affect only a subset of users, or all users. This will be scored between 0 and 3.

# Configuration Management Plan

# The overall objective of a configuration management plan is to document and inform fellow project members about configuration management with the project, what tools will be used and how they will be applied in the Project. The Configuration Management plan defines the project structure and methods for:

* Identifying, defining, and baselining configuration items (CI)
* Controlling modifications and releases of CIs
* Reporting and recording status of CIs and any requested modifications
* Ensuring completeness, consistency, and correctness of CIs
* Controlling storage, handling, and delivery of the CIs

The intended audience of the Configuration Management Plan is the project leader or any other senior leaders in the group whose support is needed to carry out communication plans.

## Configuration items and tools

* + 1. We will be using the Android SDK
    2. We will be using the Android Developer Toolkit, based in eclipse
    3. We will be targeting Android 4.3
    4. We will be targeting the tablet and phone form factors

## **Change management and branch management**

* + 1. We will be using GitHub as our repository
    2. We will use GItHub to manage branches
    3. Changes will be fielded via Pivotal Tracker

## Code commit guidelines

i. Don’t ignore Exceptions

ii. Don’t catch generic Exceptions

iii. Don’t use finalizers

iv. Fully Qualify Imports

v. Use standard comments

vi. Write short methods

vii. Define fields in standard places

viii. Limit Variable scope

ix.Use spaces for indentation and use standard barce styles

x. Follow field Naming conventions

xi. Use TODO comments

xii Limit the line length