**Chore Change**

A picture containing room

Description automatically generated

Company: Big Ike Inc.

Submitted By: Isaac Medlin

Submitted To: Gary Clark

Email: [Isaac.medlin@oit.edu](mailto:Isaac.medlin@oit.edu)

Date: Tuesday, May 12, 2020

Version: 1.0

# Legal Notice

All product, support, services, information, and software in this app are provided “as in” without warranty of any kind, express or implied, including, but not limited to, the implied warranties of fitness for a particular purpose, and non-infringement, in no event shall Big Ike Inc. or other contributors be liable to you or any third party for any direct or indirect, special, incidental, or consequential damage in connection with or arising out errors, omissions, delays, or other cause of action that may be attributed to your use of any product, support, services, information ort software from this app, including, but not limited to, lost profits ort lost data, even if Big Ike Inc. or other contributors had been advised of the possibility of such damage.

# Copyright Notice

**Copyright © 2019 Big Ike Inc. All Rights Reserved**.

All documents, questions, references, images, audio, programs, source code or other materials whatsoever contained herein, or obtained through the Big Ike Inc. mobile applications are protected by copyright. Except as explicitly permitted below, these materials may not be reproduced in whole or in part, in any form or by any means, including photocopy, electronic storage and retrieval, or translation into any other language without the express written consent of the copyright holders.

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Author | Company | Version | Date | comments |
| Isaac Medlin | Big Ike Inc. | 1.0 | 5/12/2020 | Initial Draft |
| Isaac Medlin | Big Ike Inc. | 1.1 | 1/6/2021 | Slight adjustment to functional requirements to make planning easier |

# Signatory Page

This document was accepted by:

X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature (Gary Clark) Date

This document was submitted by:

X\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature (Isaac Medlin) Date

Table of Contents

[Legal Notice ii](#_Toc40202597)

[Copyright Notice ii](#_Toc40202598)

[Revision History iii](#_Toc40202599)

[Signatory Page iii](#_Toc40202600)

[1.1 Purpose 1](#_Toc40202601)

[1.2 Scope 1](#_Toc40202602)

[1.3 Intended Audience 1](#_Toc40202603)

[2. Project Management 1](#_Toc40202604)

[2.1 Change Management Procedure 1](#_Toc40202605)

[2.1.1 CAT Team 1](#_Toc40202606)

[2.1.2 Medium 1](#_Toc40202607)

[2.1.3 Protocol 2](#_Toc40202608)

[2.1.4 Impact Analysis 2](#_Toc40202609)

[2.1.5 Archive 2](#_Toc40202610)

[2.2 Software Delivery 2](#_Toc40202611)

[2.3 Installation 2](#_Toc40202612)

[2.4 Acceptance Criteria 2](#_Toc40202613)

[2.5 Documentation and Online Help 2](#_Toc40202614)

[2.6 Product Risks 3](#_Toc40202615)

[2.7 Customer Responsibilities 3](#_Toc40202616)

[2.8 Status Reporting 3](#_Toc40202617)

[3. System General Requirements 3](#_Toc40202618)

[3.1 Project Summary 3](#_Toc40202619)

[3.2 Relation to Existing Systems 4](#_Toc40202620)

[3.3 Hardware Platform Description 5](#_Toc40202621)

[3.3.1 Hardware requirements 5](#_Toc40202622)

[3.4 Software platform description 5](#_Toc40202623)

[3.4.1 Mobile App Software Platform Description 5](#_Toc40202624)

[3.5 Third Part Libraries 5](#_Toc40202625)

[4. Product Requirements 5](#_Toc40202626)

[4.1 Functional 5](#_Toc40202627)

[4.2 Performance 6](#_Toc40202628)

[4.2 Data Transfer Description 7](#_Toc40202629)

[4.3 Security/Safety/Constraints 7](#_Toc40202630)

[5. User Profiles 7](#_Toc40202631)

[6. Glossary 7](#_Toc40202632)

[7. Appendices 8](#_Toc40202633)

[7.1 Change Request Form 8](#_Toc40202634)

1. Introduction

## Purpose

The purpose of this document is to propose a design and implementation of \*app name\*. The rest of this document is responsible for defining the document’s format and the proposed project design. Upon acceptance, the proposal will be a living document which will guide the development and implementation and maintenance processes.

## Scope

Information regarding the project management, a general description of the system, critical product requirements and discussion on user profiles are all in scope for this proposal. Out of scope for this proposal are billable rate, work estimate, and any other financial/monetary impact.

## Intended Audience

This document is intended for Gary Clark, and any other interested party not listed.

# 2. Project Management

## 2.1 Change Management Procedure

### 2.1.1 CAT Team

As the project develops and any changes are identified as necessary to project completion, the Change Acceptance Team (CAT) will be notified about the request. The entirety of the CAT Team will be comprised of Gary Clark, \*etc.\*. Change submissions will be reviewed by the team and a determination will be made to accept or reject the request, including some reasoning which led to the decision. All requests will be archived by Isaac Medlin.

### 2.1.2 Medium

All changes must be submitted by a member of the CAT Team and must conform to the change Request Form (see Appendix).

### 2.1.3 Protocol

All changes will be processed and decided within 7 business days from submission. Upon acceptance, a member of the CAT Team will notify the requestor of the resulting decision via the email provided on the Change Request Form.

### 2.1.4 Impact Analysis

In order to retain feasibility, all change requests will analyze impacts, especially to time. If changes submitted put the project at risk of being infeasible to complete within the established development timeline, it will be rejected regardless of any other factors.

### 2.1.5 Archive

All change requests will be archived for reference and will be saved at least through project completion in a google drive as a jpeg or pdf. Archived change submissions will be available upon request to Isaac Medlin.

## 2.2 Software Delivery

Completed software will be distributed to Gary Clark no later than the end of winter term. Delivery will include a link to the downloadable application from the Google Play Store, and written documentation detailing application usage.

## 2.3 Installation

Installation will consist of an application download from the Google Play Store.

## 2.4 Acceptance Criteria

Acceptance criteria will be determined by Gary Clark, based on the functionality and completeness of the project as defined in this document.

## 2.5 Documentation and Online Help

No online help will be created for this work effort. All documentation will be written and will detail system requirements, startup procedure and information regarding operation.

## 2.6 Product Risks

The risks for this product are based on time considerations, such as learning new technology and identifying 3rd party libraries.

## 2.7 Customer Responsibilities

The customer will be responsible for providing an Android Smart Phone that adheres to the system requirements.

## 2.8 Status Reporting

Status reports will be sent by Isaac Medlin to Gary Clark weekly, on a day agreed upon by all members of the CAT team. The status report will include a list of work which is scheduled to be to be worked on in the upcoming week. If any issues have been identified in the previous week, these will also be included in the status report, along with tried alternatives and suggested solutions.

# 3. System General Requirements

## 3.1 Project Summary

This app idea came from some good friends, they just had their first kid and want a way to be able to reward her for doing chores around the house. They want an app that can help them give their kid an allowance in a fair and productive way. This would require the creation of a mobile app. The main idea of the app is for a parent account to create chores and assign a dollar value to each chore that needs to be done. The child account then would be able to complete the listed chores. Once completed, the parent account would get a notification, verify that the jobs done and either accept/reject the job. If accepted, the dollar value from the completed chore would be put into a running total on the child’s account. When the child wants to cash out that amount, the parent account would get a notification that the child wants their allowance, with the amount they owe them. This app would in total have 3 different GUIS: A parent GUI, a child GUI, and a combined GUI.

The parent GUI will have 3 main parts: A chore creator, a “children” tab and an account page. The chore creator will have a list of chores under a to be done section, in progress and a list of chores pending the parent’s approval. To create a chore there will be a little plus icon that will pull up a simple form that the parent will fill out. The form will have a set amount of inputs: Name, description, picture (optional) and the dollar amount upon completion. Once created the chore will be added to the to be done section of the chore creator tab. The children tab will have all the accounts listed that are under the parent account. It will provide the accounts names, and the current running total for each account so that the parent can be financially prepared for a chore cash out at any moment. The account page will have the normal things, a change password, change name, info, settings, and a join code. The join code will be the most useful part of the account page. The join code will be what allows a child to join under a parent and is a must for the entire app to work.

The child GUI, upon creation will need a name and a join code to link up with a parent account. The child GUI will have 3 main sections also, their bank, chore list, and account. The account page will just be a simplified version of the parents account page, with change name, change password and settings. The bank tab will have the amount of money the child’s earned with the chores. It will have a “cash out” button that will send a notification to the parents account. This will put the transaction into a “pending” mode that the child must confirm once the parent gives them the allowance. The chore list section will have a list of all chores with the dollar amounts that the parent has created, a list of chores in progress and a list of “pending approval” completed chores. Upon clicking on the chore, it’d pull up the name, description, picture and dollar amount. Within this chore there will be a start button, which will move it to the parents and child’s in progress section of their chore GUI. This will add a level of competitiveness if there’s multiple children under a parent account, who can get to the best chore the quickest. Once the kid gets the chore done, they’d reopen the chore they started and click a complete button. This in turn would add it to the “appending approval” section of the chore list. If accepted by the parent, the child would get rewarded the dollar amount, while the chore disappears. If rejected by the parent, the chore would be moved back into the in-progress section until completed properly.

The combined GUI will be able to take both GUI’s and mash them into one. This is for parents with younger children who won’t have a phone yet. The concept and design each GUI remains the same, but there would be a hamburger button that pulls up a list of accounts. Upon clicking a different account, the app would switch between the current account and the selected account, while hot-swapping GUI’s (if switching between parent and child).

I will be using a SQL database for this project, since there’s a set number of columns for each chore, it shouldn’t be to big of a challenge to send and receive data.

The main stretch goal I’d want to complete is a roommate mode: instead of paying out money to a child, it’d assign chores to each roommate fairly. Each roommate would be able to add chores that need to be done to the app, and once it hit the next day, everyone would randomly get assigned the same number of chores. For example, for 3 roommates that added 6 chores in a day, each would wake up to 2 chores to do each the next day. This would stop disgruntled roommates from having to do all the chores while maintaining a certain amount of “fairness”. From what I hear and see, this would benefit multiple houses and people, and since it doesn’t deviate from the main app idea too much, it wouldn’t be an unreasonable goal to get done.

## 3.2 Relation to Existing Systems

This project has no relation to other systems, I’d be building my app and database structure from the ground up.

## 3.3 Hardware Platform Description

### 3.3.1 Hardware requirements

Minimum android version of 5.0, Lollipop

150 MB of free memory

Wi-Fi capabilities are a must

## 3.4 Software platform description

### 3.4.1 Mobile App Software Platform Description

Ability to render pictures coming from a database

Ability to access the internet

## 3.5 Third Part Libraries

Nothing that I can think of at this moment, might be updated while implementing

# 4. Product Requirements

## 4.1 Functional

Parent/Child Accounts

1. Users shall be able to download the app through the androids Google play store
2. Users must be able to register and log in with two different kind of accounts:
   1. Parent account
   2. Child account
3. Parent account must be able to post chores
   1. Multiple child accounts can be linked to one parent account
   2. Can view all the children’s accounts banks when their linked to the parent account
      1. Ensures that when the children want to be paid out, the parent will have enough money set aside for them
4. Children accounts will be able to access the posted chores
   1. Each child account will have a bank, tallying up the total cost of chores they’ve done so far.
   2. Children accounts will have a payout option, resetting their bank back to $0.00 and notifying the parent account that the kid requested the payout amount
5. Each account will have an account page with:
   1. Name: which can be edited
   2. Password: changeable through a security question
   3. Parent account only:
      1. Manage which children accounts connected to them
   4. Switch account: allows for multiple sign ins on one device
6. A single phone will be able to log into both accounts simultaneously
   1. For younger children who may not have a phone yet
   2. Account swapping for easy navigation between them
7. Chores will have 4 categories they can fit into:
   1. To do: recently posted chores, waiting for a child to accept it
   2. In progress: One of the children accepted the chore, and is in the progress of completing it
   3. Awaiting approval: Marks that the child is done with the chore, after checking for completion the parent will be able to accept that the chore’s completed
      1. Will alert the parent account when a chore is awaiting their approval
   4. Completed: Shows all the chores completed within the last 7 days
8. Chores will have a payout
   1. Completely set up by the parent
   2. Once completed it’ll add the total to the child’s bank
9. Chore creation will include the following input fields:
   1. Chore Name
   2. Description
   3. Payout amount
   4. Picture
      1. Pictures will be interfaced through the Samsung photo gallery with camera support
10. Child and Parent GUI’s will be slightly different:
    1. Child GUIs:
       1. Only be able see and accept chores
       2. See their bank amount with a cash out option
       3. Settings page
    2. Parent GUIs:
       1. Being able to post a chore
       2. See all posted chores and their status
       3. Page with all the names of their linked children accounts and their bank amount
       4. Notifications tab that notifies them when specific things happen:
          1. When a child requests a cash out
          2. When a child completes a chore and requests approval
       5. Settings page
    3. Refresh button on both accounts to call to the database update changes
11. Database Code
    1. Tables for chores and each type of accounts.
    2. Sanitation and encryption
    3. Database worker that deletes entries after 7 days
12. Networking Code
    1. Chat room for the whole family to discuss stuff

\*\*Stretch Goal\*\* Roommate Accounts

1. Same idea as the other accounts with a little bit of a difference.
2. Each account will be able to create a “house”
   1. From there other accounts will be able to join in of the house
3. Each account will be able to post chores
4. Chores will have no payout
5. At midnight each night (Server time), the chores will be evenly divided between roommates in a random order.
6. Same Account page as parent accounts

## 4.2 Performance

The app will load all the pages within a second, delays will be due to database calls. Database calls to retrieve information will happen on app load-up. Database calls to update information will happen whenever somethings updated in app.

## 4.2 Data Transfer Description

Data will transfer at a minimum of 5 Mbps from the server to the users.

## 4.3 Security/Safety/Constraints

Passwords and security answers will be hashed and salted using BCrypt. The database will have sanitation inputs to prevent SQL injections and other ill intent inputs.

# 5. User Profiles

ChoreChange has two core user profiles. The parent account will be able to host multiple child accounts and the ability to create chores. The child account will have the ability to accept and complete chores to earn money.

# 6. Glossary

BCrypt- A password hashing library.

CAT Team- Change Acceptance Team.

SQL- Structured Query Language, a language used for managing relational databases.

Google Play Store- Android mobile operating system’s marketplace for apps.

Bank- The accruing total of the earned money for child accounts.

“Cash out”- An option child account gets, notifies the parents that they want to get their allowance they’ve earned. Resets their bank to $0.

# Appendices

## 7.1 Change Request Form

A screenshot of a cell phone

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **Change Board Approval:** | | |
| **Name** | **Signature** | **Date** |
| Isaac Medlin |  |  |
| Calvin Caldwell |  |  |