
# Design Document

Zach Cornell

Caleb Stevenson

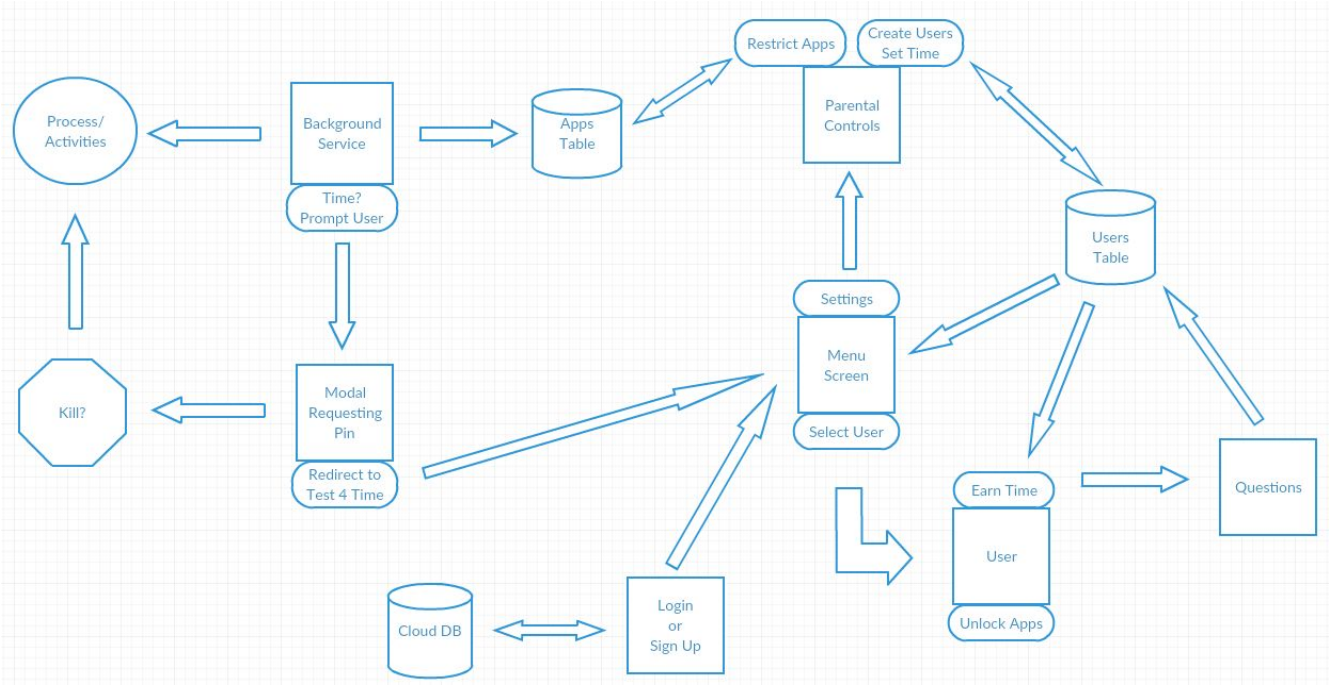
Carson Schaefer

**Languages**

To develop the system for the Test4Time app, the team will primarily be using the Java programming language. The team will also be using SQLite for any back end database and storage options the group might need. SQLite will be used to execute CRUD tasks for data stored on the device. The Java programming language is ideal for this application because it is going to run on android devices. XML files will need to be modified when constructing the user interface.

**Framework, Libraries, and APIs**

The primary library and API the team will be using is the Java Android Development Kit. This will assist us in creating an application to function on devices running the Android operating system. It will also provide efficient ways for displaying images on device screens and for implementing necessary information storage options. The team has some experience in developing Android applications as well as iOS applications.




**Basic System Organization**

One aspect of the Test4Time application will involve a database configuration to access stored information regarding the user's current status. This will involving storing and retrieving information about the child's current user profile, grade level, and current time earned. The database configuration will also make use of retrieving information for parental control settings and options. This information will be stored in a SQlite database on the android device, as shown in the figure above with Apps Table, Users Table.

USERS:

| Attribute | Data Type | Null? |
|---|---|---|
| ID | INTEGER | AUTOINCREMENT |
| NAME | TEXT | NOT NULL |
| TYPE | INTEGER | NOT NULL |
| PIN | INTEGER | NULL |
| GRADE | INTEGER | NOT NULL |
| TIMEUP | INTEGER | NULL |

APPS

| Attribute | Data Type | Null? |
|---|---|---|
| ID | INTEGER | AUTOINCREMENT |
| APPNAME | TEXT | NOT NULL |
| PACKAGE | TEXT | NOT NULL |
| PROCESS | TEXT | NOT NULL |

A database helper API has been written to perform CRUD tasks on this information.

This application, besides functioning as an environment for users to answer math questions based on their grade level, will also run as a service monitoring the devices services. This service, will check which applications are running on the device. When an application gets launched to the foreground, the Test4Time service recognizes this. The user will be redirected to a password prompt if no time has been acquired as illustrated by the Background Service section in the figure above. This will allow the parents to enter their pin to unlock the tablet and use the requested application. If the child is trying to access a blocked app, then they can choose to go answer more questions to earn time. This will launch the Menu Screen of the

Test4Time application where the child can select their user account. As the child answers more questions, the database will be updated with their current amount of time.

The process that is requested is not killed. (The android kernel does not let our activity kill processes we don't own) Users are always redirected to the Test4Time pin activity.

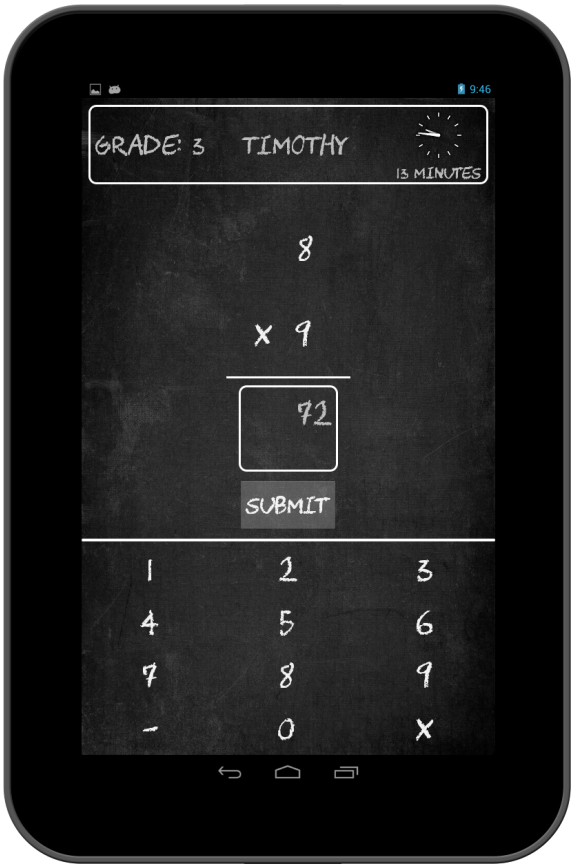
The Test4Time application requires the following permissions to perform these tasks. First, it needs the GET_TASKS permission to access what applications are running or trying to be opened. The other permission required is RECEIVE_BOOT_COMPLETED, which triggers our background service to start when the phone is turned on.

**User Interface**

The visual display tool in Android Studio will provide the basic tools for constructing the user interface. XML files will need to be written to create customs widgets and pieces to be displayed.

The figure below is the current user interface the child will view while answering math questions. The team simplified the UI so the child does not need to have a keyboard pop-up so they can type in an answer. The team created a custom keypad at the bottom of the screen so the child can easily press the numbers they want to enter. Currently the grade level, child's name, and current time is displayed in a heading section at the top of the screen. There is also a submit button in the middle of the screen for the child user to press when they want to check their answer. It has been considered whether their answer should be automatically submitted, but this is yet to be decided. The layout is currently locked in the a portrait orientation, as requested by the client. This is more consistent with the vertical math problems children solve.

The team has started working on the main start menu of the application as well. This will be shown when the user first opens the Test4Time application. A list of child users will be displayed where the child will select their name and be sent to the math question page to earn time. There is also a settings button which will open the Parent Menu. In the parent menu will again be a list of child users. However, the parent will be able to update the information of the child accounts, specifically their grade level as they get older. The parent will also be able to add accounts for their children from this page. Further, in this parent menu will be a section to display the list of installed applications on the device. The parent will be able to view and check which applications they would like blocked for their child when the child does not have enough time built up. Access to the parent menu will require a password so that children will not have the access to these activities.


9:46
GRADE: 3
TIMOTHY
13 MINUTES
8
X 9
72
SUBMIT
1
2
3
4
5
6
7
8
9
-
0
X


10:33 PM
Test4Time
USERS
User 1
User 2
User 3
User 4
User 5
User 6
User 7
User 8
User 9
User 10
User 11
User 12
User 13
User 14
User 15
User 16
User 17
TEST 4 TIME
TEST 4 TIME

**Design Methodologies**

The way Test4Time is approaching this project is building a solid foundation and then working from there to add features as necessary. The team will give weekly updates to the client showing our progress and to consistently check if Test4Time is headed in the right direction based on the client's specifications. Using the Agile methodology the team is able to keep themselves aligned with the specifications and make sure they are giving the client exactly what he wants. If any changes need to be made it will always result in a quick fix as the team has checked with the client each step of the way.

**Testing**

This will be difficult due to the variety of android devices. Blocking apps worked on our tablets and phone running Android 5.0 and earlier but did not run on a phone running 5.1. Encountering these fail situations will be different with a limited number of devices. The IntentService was tested to make sure it is always running. This includes device booting and when the Test4Time application is killed. Since the requested application is only moved out of the foreground, apps like Pandora radio can start to play music and appear in the notification bar if the intent service doesn't start the Test4Time pin activity fast enough.

The client has plans for testing the Test4Time application with real users once the project is closer to completion. This will include providing sample parent users access to a testing version of the application so they can view how children will interact with the app.

**Gantt Chart**