Deliverable 1

SEG2105

Group Members:

Christopher Francis (8661478)

Kevin Power-Peacock(8586032)

Benjamin Vokey(8649554)

Thomas Scott Fulton(8765062)

Deliverable 1: Requirements and Use Cases

* List of at least 20 functional requirements and 5 non-functional requirements
* 3 fully described use cases (no need for full diagram though)

Dividing work:

* Each of us do 5 functional requirements
* 3 of us complete 1 use case. Remaining person do the other 5 non-functional requirements

Writing Use Cases (Paraphrased from the textbook)

1. Name: Short verb-phrase describing action user will do to system
2. Actors: List of users or actors who can perform this use case
3. Goals: What the use case and actors are trying to achieve
4. Preconditions: Describe state of the system before use case occurs, list any conditions that must be true for it to occur
5. Summary: Summarize what occurs as the use case is performed
6. Related use cases: use cases related to this case
7. Steps: Table format showing steps user takes and related system operations
8. Postconditions: State of the system after the use case has occurred

Kevin’s Work

**Name:** Use Case #1, Complete Chore

**Actors:** Children, Parents

**Goals:** Child wishes to affirm that a chore has been completed.

**Preconditions:** Before the case can be fully completed by the child, the parents must *confirm* that the chore actually has been done. *Confirmation* can occur before or after a child has *completed* a chore.

**Summary:** The children will *complete* their chores with this case. After finishing the physical task, they shall denote the task as “completed”. If the chore has been *confirmed* by parent actors, then the noted reward points will be given to the child. Otherwise a *confirmation* request will be sent for parents to review at a later time, which may be accepted or denied according to the parent’s decision. Accepted *confirmations* shall award their points, while denied ones will  be returned to the task list.

**Related Use Cases:**

* Confirm chore (if parents have not confirmed the chore, the chore can not be completed by the child).
* Reward Points (points are to be awarded upon completion of chores).

**Steps:**

|  |  |
| --- | --- |
| **Child** | **System** |
| 1. Child requests that a chore has been completed          6) Chore points awarded to child should the chore have been completed up to parent’s standards. | 2) System checks whether or not the chore has been *confirmed*  3) If the chore has been confirmed, child is rewarded.  4) Otherwise a pending request is sent to parent actors to *confirm* the request  5) Chore points are awarded to child upon affirmative *confirmation* of chore. Otherwise the chore is to return to the original chore list in the event it has not been completed up to parent’s standards. |

**Postconditions:** If the parents have already *confirmed*\* mentioned chore, the tasks is deemed complete and popped from the task list. Otherwise a request is sent to parent actors for determine whether or not the chore has actually been completed

Functional

1. The tasks in household chore manager shall have a numeric point reward between 1 to 10 to be awarded to the child(ren) who complete the chore.
2. The chore manager’s numeric reward points shall be assigned by the parent actors.
3. Parent actors in the household manager will be notified of pending *chore completion* requests upon first starting up the app.
4. The household manager points will be stored on the same location responsible for storing instance information regarding children actors.
5. The household manager will provide an interface for parents to assign rewards to be bought using chore points (please note that it is the parent actors’ responsibility to actually implement the reward)
6. The household manager’s aforementioned reward interface will be on a read-only basis for children actors.

Chris’ Work.

Functional

1. The tasks in household chore manager will have the option to add a due date, urgency(1-5), and add notes.
2. Each account will have a name, profile picture, and birthday.
3. Household chore manager will have the option to assign an account to each chore.
4. The system shall hide the chore from list when completed.
5. The system shall have a completed chore section.

Non Functional

1. Household chore manager will support version 25.3.1 and up.
2. Household chore manager's database shall be Firebase.
3. Household chore manager will be build in Android studio version 2.3.3.
4. Household chore manager will be programmed in Java Version 8.
5. The manager shall meet or exceed 99 % uptime.

Ben’s Work

**Name:** Use Case #2, Create Chore

**Actors:** Parents

**Goals:** Parent wants to create a chore for the children to complete

**Preconditions:** There must first be children accounts to assign the chores to before the task can be started

**Summary:** The Parent will select the create chore option of the program, and be brought through several steps from the name, description, point value, and children assigned. Then the system will assign that chore to the selected children

**Related Use Cases:**

|  |  |
| --- | --- |
| **Parent** | **System** |
| 1. Parent selects the create chore button   3) Parent inputs values for the fields and selects finish button  5) Parent selects from the list of children and confirms the chore creation | 2) System brings up a screen with editable text fields showing “Chore Name”, “Chore Description”, and a number chooser with the name “Chore Points” as well as a button that says “Finish” or some other word  4) System brings up pop-up menu with a list of children to assign the task to, wherein the parent selects the children they want to give the task to  6) System places the chore in the “To be completed” section of the child’s app screen |

**Postconditions:** The parent’s screen returns to normal except for an additional chore to be marked as confirmed. The child’s screen has a new chore listed under the to be completed section.

**Functional**

-There will be two account types, Parent and Child

-Parent accounts will be able to assign chores and modify their contents

-Child accounts are only able to mark them as complete

-Parent accounts will be able to reassign completed chores

-Chores will be able to be assigned to one or multiple child accounts

**Scotts Work:**

**Name:** Use Case #3, Create Account

**Actors:** People

**Goals:** A Person wants to create an account for the chore manager

**Preconditions:** None

**Summary:** A Person (Child or Parent) will create an account with this case. The person will select the create account option and then proceed to fill out the required information sections; their name, their birthday and a profile picture. The account must be designated as either a parent or child account.

**Related Use Cases:**

None

**Steps:**

|  |  |
| --- | --- |
| **Child** | **System** |
| 1) Create Account option is selected  3) Person inputs values for the fields and clicks the "Next" button.  5) Person selects account type, and presses the "Finish" button. | 2) System brings up a screen with input fields for "Name", "Birthday", "Profile Picture" as well as a button that says "Next" and one that says "Cancel".  4) The System then brings up a field with two options to choose what type of account shall be created. The options are "Parent" or "Child". There is also a button that says "Finish".  6) System creates the account and stores the values to memory. |

**Postconditions:** If all fields have information typed into them, an account is created with the information provided. If invalid data is entered or the user selects to cancel, no account is created.

**Functional**

-Child  
actors will be notified of new chores assigned to them

-Child  
actors will be notified of chores “due” on the date of usage

-Chores will have the option to be sorted by Due Date or by Urgency

-Chores  
will have the option to be saved for future use, to be reassigned

-Parents will have the ability to delete saved chores