# ”What was I meant to do again” (Exploration of event boundary on the failure of prospective memory)

*Aldy Syahdeini*

Master of Science

School of Informatics
University of Edinburgh

2017

# Abstract

This doctoral thesis will present the results of my work into the reanimation of lifeless human tissues.

# Acknowledgements

Many thanks to my mummy for the numerous packed lunches; and of course to Igor, my faithful lab assistant.

# Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Aldy Syahdeini*)

# Table of Contents

# Chapter 1

# Introduction

## 1.1 Prospective Memory error

Have you experienced when you wake up from your bed in the morning, put your glasses on and go to the kitchen to get a glass of milk. But when you are in the kitchen, you totally forget what you intended to do. This phenomenon is called prospective memories failures.

Prospective memory is the ability in the future to remember to do an action that previously planned without being instructed to do so (Groot et al., 2002). This type of memory is different with retrospective memory which is the memory that we use when we are answering a question in the exam. Retrospective memory involves remembering event, words, and so on from the past typically when deliberating to do so.

Prospective memory failures are common in everyday life, almost 50% of forgetting in our daily routines are due to of prospective memory error (Crovitz and Daniel, 1984). This memory failure can lead to embarrassment such as forget that you had arranged a meeting with your friend and even result in serious injury or death. One example of a horrible case is ”After a change in his usual routine; an adoring father forgets to turn toward the daycare center and instead drove his usual route to work at the university. Several hours later, his infant son, who had been quietly asleep in the back seat, was dead ” (Einstein and McDaniel, 2005). So it is important to have a great understanding about prospective memory error.

But what makes us forget ?. Radvansky and Copeland (2006) and Radvansky et al. (2010) shows that if people make a transition from one event to another, for examples move from one room to another room, they tend to forget more information than if they do not. Cockburn and Smith (1994) Show that stress and anxiety cause us to

become absent-minded and thus produce failures of prospective memory. There is also a lot of study about ageing and its relation to prospective memory, one of it is a study conducted by Scullin et al. (2012) found older people tend to make more error than younger people on a prospective memory test.

The purpose of this MSc Dissertation project is to build an application that can use to conduct an experiment about prospective memory error, analyse the effect of multiple intention on a failure of prospective memory and to make a further understanding of what happens during event boundary (e.g., moving to another application inside the smartphone) by tracking the activity of the participant during the prospective memory task. The experiment conducted on this thesis is originally based on studies done by Lisa M. Stevenson & Richard A. Carlson (what Did I Come here to do ?, Pennsylvania State University 2016).

## 1.2 Project goals

The main goals of the thesis are to create an application that can be used to other researchers to conduct a prospective memory experiment. The application should able to conduct three type of studies from Prof. Alan Carlson's experiment. Also to conduct an experiment in how people lost their intention on the event boundary and the effect of multiple intentions on prospective memory error.

## 1.3 Structure of dissertation

The document is structured as follows

Jesus this need to be changed

- In the **Literature Review** chapter provide an introduction about prospective memory
- In the **Experiment Design** chapter provide an information about the architecture and the design of the application. The main flow of the application and the front end looks is explained. Also, the experiment design and properties is also provided.
- In the **Implementation** chapter provide information about the technical implementation of the experiment application based on the design and the require-

ment. This chapter explain how the main flow of the application works and how the features is implemented.

- In the **Experiment result** chapter provide the result and analysis of the result of the experiment.
- in the **Conclusion and Future Work** highlight the summary and achievement of the application and experiment. And also giving an opinion about possible future improvement and research.

# Chapter 2

# Literature Review

## 2.1 Prospective memory and retrospective memory

Tasks such as buying milk in a supermarket on the way to work action, turning off the oven and taking a medication are categorized as a prospective memory task. Prospective memory is used constantly in everyday activity (Wilkins and Baddeley, 1978), (Winograd, 1991). There are a lot of definition about prospective memory, but generally a prospective memory is defined as remembering to carry out planned actions at a particular time in the future without being instructed to do so (McDaniel and Einstein, 2007); (Groot et al., 2002). While task such as answering the question on an exam or remembering the person name on the party is categorized as a retrospective memory task. Retrospective memory involves remembering events, words, and so on from the past typically when deliberating trying to do so.

According to Baddeley and Wilkins (1983), it's very hard to differentiate between prospective memory and retrospective memory because there is no clear cut between them. for example, To remember to call your father, you should able to recall his number and how to use the phone, and not call him while he watches a football match. Brandimonte et al. (1996) call this as retrospective component of a prospective memory task. Cockburn (1995) stated that content of the information is similar to both memory type but the essential difference is prospective memory require memory for intention and the cue for retrieval has to be self-initiated. Guynn et al. (1998) also state that retrospective memory is driven by low information content while retrospective memory is driven by high perceptual information, such as question during an exam.

Furthermore, Remembering only the retrospective memory component of a prospective memory task will not produce successful prospective memory. In fact, numer-

ous prospective memory failures happened because the failure of remembering the prospective memory component (Einstein et al., 1992). Interestingly, the component of retrospective memory sometimes forgotten in a simple prospective memory task, for instance when we walk to the kitchen and sometimes forget what we are intended to do there (Brandimonte et al., 1996).

## 2.2 Cognitive process of prospective memory

Some researcher believes that prospective memory proceeds through encoding, retention, retrieval, execution and evaluation phase. According to Ellis (1996) In the Encoding phase, the *when*(retrieval criterion), *what*(action to be performed) and *that*(intent or decision to act) are encoded. Then this intention representation must be retained until the opportunity to fulfill the intention occurs. this delayed can vary from a second to a week. Einstein and Mcdaniel (1990) categorize retrieval process, event-based or time-based retrieval. On the event based retrieval, the retrieval happens if there is a particular event or physical stimulus that associated with the intention. for example telling a message when you meet your college. On the other hand, time-based retrieval require execution of action after a certain time (Ellis, 1996); (Mcgann et al., 2002). Therefore, successful prospective remembering can be described as a process that support the actualization of delayed attention and the associated action, and it is strongly as associated with control or coordination of future action (Ellis, 1996).

## 2.3 prospective memory error

Prospective memory error is defined as a failure to do a planned action at some point or at a particular event in the future. Kliegel and Martin (2003) state that prospective memory failure is the most frequent memory failure in everyday life. The ability to remember the planned action is a critical factor in human functioning. The consequence of a failure of prospective memory can be trivial for example forgetting to buy some milk on the way home from work. But it can also have severe consequences, for example, the doctor forget to took the scalpel from his patient after an operation. In fact, Shorrock (2005) reported that 38% of accidents on the traffic controllers in the UK due to memory error involves the failure of prospective memory.

Many researchers has different view on the prospective memory error and what cause it to happen.

Kvavilashvili and Ellis (1996) try to categorize and differentiate a various kind of memory error with a prospective memory error. They claim that *action-slip*(Heckhausen and Beckmann, 1990), *actions-not-as-planned* (Reason, 1979) and *absent-minded error* (Cohen and Conway, 2008) should not constitute as a prospective memory error. These errors happens because the failure that occurs during the execution or performance of the intended action. for example, in absent minded error people lose the context of an intention and carry out an unintended action instead of the intended one. In contrast, prospective memory is focused on the failure to retrieve intended action. While Guez and Naveh-Benjamin (2013) argue that these type of error should be considered as part of prospective memory error because prospective memory contains some element of retrospective memory such as the context of intention. Moreover, Reason (1985) explained further on how the element of memory; context, intention and attention influence prospective memory error. In addition, Cockburn and Smith (1994) argued that stress and anxiety make a person to experience absent minded error hence make a prospective memory error. and Scullin et al. (2012) found older people tend to make more error than younger people on a prospective memory test.

## 2.4 Prospective memory and intention

Because prospective memory refers to remembering intentions so it would be better to have a good understanding of intention first. For example to understand the nature of intention and its phenomena,the category of intention and how it related to everyday activities and what happen to intention during prospective memory error. The explanation of these question maybe gives us more understanding about the correlation between intention and prospective memory error.

Kvavilashvili and Ellis (1996), Gauld and Shotter (1977) define an intention as a person's readiness to act in a certain way in the future. what has to be done and when to be done should be defined clearly.

Searle (1983) distinguished intention into two types, prior-intention and intention-in-action. A prior intention is an intention that is defined prior to action, while intention-in-action is a spontaneous action, for example going to the toilet when you need to urinate. A prior intention is always occurred as a result of conscious decision to act in a certain way (Heckhausen and Kuhl, 1985). Furthermore, Gauld and Shotter (1977) categorized prior intention into two categories, delayed intentions and immediate intention.The delayed intention is a postponed intention that will be executed at some

point in the future, and when a person begins to carry out their prior intention immediately after a decision has been made or after they see a particular cue for the intention.

The difficulty of retrieval of the delayed intention make persons miss the prearranged moment or cues, and this make people fail to remember. Even though people able to retrieve the delayed intention, but when the intention is initiated and transformed into an immediate intention, people can still lose their intention and prospective memory error occurs. Furthermore, Reason (1985) explain how a change in the intention make people experience memory error by categorizing two phenomena called *detached intention* and *lost intention*.

#### 2.4.0.1 Detached intention

Detached intention happen if the content of the intention detached from its proper setting and misapplied to something other than its original intention. For example, the case when a person switches off the television instead of the oven. the possible explanation for this phenomena is because the intention is not framed completely, probably because the focus of attention was claimed on other things (this will be explained further on the attention section). Another explanation is that the intention was framed completely but has not sufficient level to be retained until the moment of execution. Another explanation is there is another combination of intention that has similar action and trigger from another object which similar kind of action is appropriate (Reason, 1985).

#### 2.4.0.2 Lost of intention

While detached intentions happened because of partial failure of the intention and retention system. Lost intention is a complete breakdown at one or more of the stage of formulation, encoding, storage, or retrieval of the intention. One typical case is when an intention is lost during the retrieval phase. for instance when a person walks into a room and become aware that he/she can't recall the original intention of the activity (Reason, 1985).

## 2.5 Prospective memory and attention

When we accidentally put our phone in the fridge instead of our food or when we pour the second kettle of water into a freshly made coffee. These slips of action frequently

occur as the result of misdirected or diminished attention Reason (1985) James defines attention as ”the taking possession by the mind, in clear and vivid form, of one out of what seem several simultaneously possible objects or trains of thought”. There is a minimum degree of attentional involvement is necessary to ensure the right execution of the sequence of attentions, and to avoid someone make a mistake due to some kind of attentional failure.

Reason (1985) define attentions as the gatekeeper of consciousness. This definition marks an important role of attention and consciousness in the performance of delayed intention on prospective memory. A person must be conscious of the plan to perform an action. To be conscious about it, the plan should be the focus of attention. the attention should be kept at the encoding phase when the action is planned and at retrieval when the action is performed.

But error can also be occur when a person is putting too much attention on the ongoing activity, for example, running down the stair two at a time, this should be an automatic activity but when a person does it with too much attention then it can be very disruptive.

Moreover, dividing attention is also assumed to reduce the contribution of a controlled process, thereby reducing performance on a memory test that involves conscious recollection (Jacoby et al., 1989). Some previous study also shows that there was a substantial reduction in prospective memory performance when attention is divided (McDaniel et al., 1998) (Guez and Naveh-Benjamin, 2013).

## 2.6 Prospective memory error at event Boundaries

We walk to the park, read a book, watch a movie and do numerous things, one after another. These stream of actions consist of events. How we split up these stream of action into events and stored them into memory influence how we think and what to remember. Memory and cognition are heavily influenced by event and how a person structures them (Radvansky, 2012). Radvansky et al. (2011) introduce an event model which is a mental model that captures the content and structure of an event that people experience.

Radvansky (2012) also suggest that when persons make a cognitive transition from one event to another, they will experience an event boundary. Such transitions can be a change in location, a causal break, the introduction of a new activity, and so on, as long as they involve a shift from one event to another. On some condition, event boundaries

can disrupt memory. When people experience event boundaries they mentally update their event model. Radvansky et al. (2010) investigate about this phenomena in the reading experiment and shows that the updating effect of a mental model increases the reading time of a sentence. the increase reflects increase on cognitive effort need for the updating.

Furthermore, Radvansky et al. (2010) found that when people pass through the doorway to move from one location to another, they forget more information that if they do not make such a shift. This effect is similar to the result from other research in text comprehension that shows that shift in location decline memory performance (Curiel and Radvansky, 2002); (Haenggi et al., 1995); (Radvansky et al., 2010); (Radvansky et al., 2003). Moreover, that study also showed that if people travelled through two doorways, they were more likely to forget than if they had travelled through only one.

Kurby and Zacks (2008) and Swallow et al. (2009) proposed event segmentation theory which explains the correlation between memory and event. The theory state that during the experience of an event, when event boundaries are identified, people segmented information into separate event models and then stored it into memory.

All these previous research result in event horizon model proposed by Radvansky (2012) The event horizon model use event segmentation theory and explained that when an event is segmented and stored as event model, it declines in availability and become deactivated. And as person experience event boundaries, a new event model is created in working memory. The active event model that is currently at the working memory is foregrounded which make it easier to retrieve ,and an available processing capacity is directed to it.

The presentation of a memory cue causes both models that contain target information to be activated this result on competition and interference, which slows down response times and increases error rates. This is why returning to a previous room does not improve memory for objects that were encountered there, and why passing through two doorways makes memory even worse than does passing through one (Radvansky et al., 2011).

## 2.7 Previous research

Lisa. M. Stevenson and Richard A. Carlson from Pennsylvania State University conducted an experiment on the failure of prospective memory. The experiment conducted

three studies, On the experiment, each participant used a mobile phone to answer eight trivia questions that randomly selected from three different topics (movies and TV, geography or Penn state trivia). On each question, an embedded link is presented, and the participant is instructed to find the answer on the web page. Subsequently, the participant is asked questions to assess their prospective memory. The experiments consist of three studies and each study answer different hypothesis.

The first study aimed to assess whether the prospective memory failure happened when a participant uses the smart phone. two question at each time are presented, and 63 participant have participated on this study. This based on a phenomenon when people clicked a link on a website, and then forget what they are looking up. The study shows that about 75% of the participant experience the failure of perspective memory which shows that the failure of prospective memory happened even when a person is using a smartphone.

The second study aimed to evaluate the effect of multiple attention with the perspective memory. The number of intention is represented as a number of questions being asked at a time. The study presents the participant with one or two questions at a time. The result shows that the failure of prospective memory more likely to happen when two questions are asked. This means that the amount of intention is an important factor of prospective memory (intentional loads).

Lastly, the third study aimed to evaluate the effect of event boundaries memory (location shifting) on prospective memory. 84 students participate in this study. One question at a time is asked, and the participant is instructed to move within a room, between rooms or stay seated. The study shows that the participants do not experience the failure or improvement of perspective memory when there has been a shift in location.

# Chapter 3

# Experiment and Application Design

This chapter describes the design of the experiment framework, both the experiment and the application side. The application will be explained from the system design point-of-view and the user experience perspective. First, The application high-level decision and work flow are explained using use case diagram. Then, the application system is describe more detail using flow chart and class diagram.

## 3.1 Experiment Design

### 3.1.1 Participant

21 Participants are participated in the study. All of the participant are student of University of Edinburgh. 3 students participate as a tester to ensure the application works perfectly. While 18 students conduct the experiment, and their data are analyzed.

### 3.1.2 Procedure

The experiment is conducted as a form of quiz where a number of questions is presented to the participant and the participant need to look the answer in the answer page. During the question and answer task there will be notification showed up. The participant can click the notification, and the android phone will get redirected to another application. After all, the participant can go back to the experiment application again.

During these experiment, three studies will be conducted. On every study 10 questions with the same category will be presented to the participant. The study is conducted using a silent lab room in a forrest hill lab and meeting room on the library.

The room is keep empty and quite to keep the participant from any distraction. These studies are explained below :

- study 1 : one question at a time will be presented to 4 participants.
- study 2 : one or two questions at a time will be presented to 11 participants.
- study 3 : one question is presented to 3 participants. on each time a participant look at the question they required to change the room. The participant is instructed to walk outside the room to the corridor or come back to the room.

### 3.1.3 Goal

The purpose of the experiment is to analyze how people lost their intention on the event boundary and what is the effect of multiple intention of the failure of prospective memory.

### 3.1.4 Question

During the experiment 10 questions is used. The question is designed as simple as possible so it does not require the participant to remember long context of the question. The particular question and answer page is chosen so that the participant should read carefully to find the answer. The questions, link to the answer and its answer are listed on the table 3.1

### 3.1.5 Demographic question

The participant answer demographic question on the end of the experiment. the list of the demographic question can be seen on the Table 3.3

### 3.1.6 Input Data

The input data use in this experiment can be seen on

put it inside the appendix

| No | Question | Answer Link | Answer |
|---|---|---|---|
| 1 | What is the original name of the titanic movie ? | https://www.simplemost.com/15-fun-facts-probably-didnt-know-titanic/ | Planet |
| 2 | In the movie ”Lord of the Rings”, How tall is Gandalf ? | https://www.phactual.com/14-fun-facts-about-the-lord-of-the-rings-the-fellowship-of-the-ring/ | Seven foot |
| 3 | How many actors played both in Game of Thrones and in the Harry Potter movies ? | http://screenrant.com/best-facts-game-of-thrones-trivia/ | 9 |
| 4 | What is the meaning of Dumbledore in the Harry Potter movies ? | http://www.teenvogue.com/gallery/harry-potter-facts | Bumbelbee |
| 5 | How many years has How i met your mother been filmed ? | https://www.phactual.com/10-fun-facts-about-how-i-met-your-mother/ | 9 years |
| 6 | How many baloons are attached to carls house in the ”UP” movie ? | https://filmschoolrejects.com/10-fun-facts-about-pixars-up-1749a61575ca/ | 10,297 |
| 7 | Where does marvel get the idea of the black spiderman suit ? | http://screenrant.com/best-marvel-facts-trivia-movies-tv-comics-superheroes/ | Fan or Randy |
| 8 | What is the most expensive movie of all time ? | https://www.factretriever.com/hollywood-movies-facts | Avatar |
| 9 | How many academy awards has the movie ”UP” been nominated to ? | http://www.imdb.com/title/tt1049413/trivia | two |
| 10 | What is the most watched episode on the show How i met your mother ? | https://ritely.com/how-i-met-your-mother-trivia/ | The finale episode |

Table 3.1: The questions used in the experiment

### 3.1.7 Consent form

Consent form is a document to formally get the approval of the participant. By ensuring the participant understand the experiment. The participant will need to sign the document. The document can be seen on

put in appendix

### 3.1.8 Participant information sheet

The participant information is given to the participant before the experiment is conducted. it has the information about the experiment and the protection of the data produce during experiment. The participant information sheet can be seen on

put inside an appendix

## 3.2 Android application and web server

The framework is originally based on the web application built by Prof. Alan Carlson and his team. The experiment framework is made extendable, dynamic and produceable so that other researcher can design and conduct various type of experiment using a high number of sample during the study of prospective memory. The Experiment framework consist of android application and web server application. The web server is used to upload the input file, set extra properties, and download the output data of the experiment , and the android application is used to conduct the experiment, track variables, and produce output data. The relation between these two component can be seen in figure 3.1

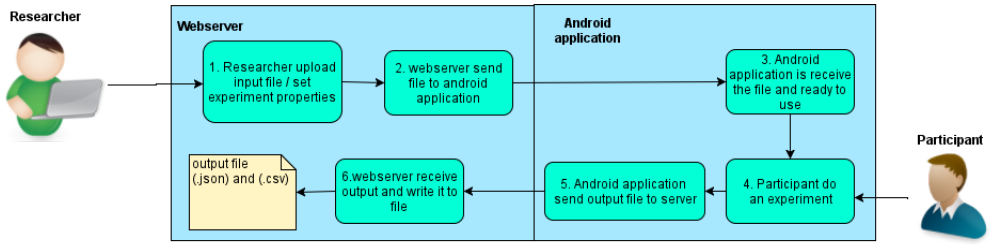


Figure 3.1: mainline of the framework

Researcher will need to start the webserver and use it to upload the input file of json format. the input file will get uploaded into the android application. On the android application the researcher can set the experiment properties for example which

experiment to conduct, the name of current participant etc. This is made because by using one input file the researcher can conduct multiple experiment with multiple participant without uploading the input file again, more of this feature will be discussed on the next implementation chapter. Furthermore, The application is able to track a group of variables of the participant during the experiment process. All the tracked variable and the mechanism of tracking will be discussed on the Tracker section. After finishing the experiment the output data inside the android application will be sent to the webserver which will be compiled to a json output file.

## 3.3 Requirement

The table 3.7 below shows the list of all requirements of the application and its description.

## 3.4 Application Design

The design of the entities, it's relationship, and the mechanism of the experiment will be discussed in this chapter. The design is based on the requirement describe earlier.

### 3.4.1 Input and output

The researcher needs to upload the input file that consist of all the experiment properties, and after the experiment done the result can be downloaded as a json file. Json (Java Script object notation) is used as an input and output format because it is very easy for a human to read and write, also for the machine to parse and generate. Most of the current programming language and analysis software support json format. The json format consist of key and value pairs, on many language it is similar to dictionary, table or struct. This input file will then be uploaded and compiled to the android application. Here is a simple example of the json format

```
{
name :"John",
age:21,
hooby:swimming
}
```

The table 3.6 shows all the field for the input and its description. The output of the application will be a json file that consist of the experiment result which consist of the

answer of the all the questions and tracked variables. Example of the input and output are shown in the appendix, show the input and output :

include the input and output example in appendix

### 3.4.2 Application entities

The input file that the researcher uploaded will be generated to an object. The architecture of the object can be seen in the figure 3.2 . Each box represent an object that consist of properties and methods. The biggest object is a Study object, this object is acted as a container for other objects.

it works like a main bone of the application that hold another object e.g experiment and category. and also control the flow of the experiment. The arrow in the figure show the pointer on which particular experiment, category, questions and notification will be presented during the experiment. The study container also act as a tracker of some variables during the experiment.

The experiment object consist of properties on how the experiment will works, e.g experiment name, number of question will be asked, and how the question will be presented. And the category object consist of multiple questions objects which consist of the question text and its answer link. Lastly, The notification consist of the parameter on when it will be shown on the experiment.

The Researcher will able to choose which experiment will be used and which notification will appeared and the participant can choose which category they want to answered. this selected experiment and category objects will be linked by study container and compiled as active category and active experiment, this will be explained much further on the implementation chapter.

### 3.4.3 Application flow

To make it easy to differentiate the experiment flow during the quiz (question and answering), the activity is divided into three parts question, answer and fill answer Activity.

Figure 3.3 shows how the application will looks like and the flow of activities of the quiz. Based on the flow of quiz activity, the application should have properties describe on the table 3.7. these properties will be used to identify the status of the activity during the experiment.

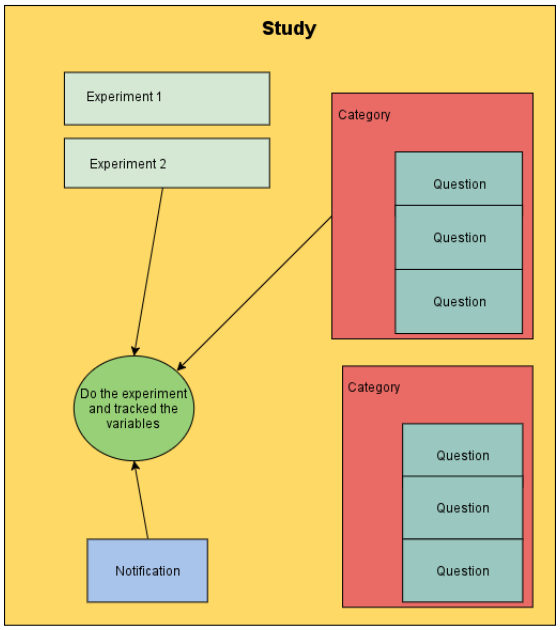


Figure 3.2: Structure of the object inside the application

To be more descriptive about the application flow, figure 3.4 shows the flow of the quiz and the how the quiz properties is updated.

- **Initialization** : firstly a *phase* is initialize. The application then check if the experiment is still going which mean there still questions need to be asked. The quiz is finished if all the question has been asked.

- **Question activity** : Secondly, the number of presented question will change randomly or stay constant. Then before the question is picked from the active category , and the questions are shown to the participant.

- **Answer activity** : Thirdly, The links for the answer page are shown to the participant, the participant then click the link and find the answer. During this phase the participant has a chance to see the question again. The participant can also decide to answer the question directly.

- **fill answer activity** is the activity when the participant need to write the answer of the question on the text box. After that, the *phase* variable is increased and the application will check again if the experiment is still going on and continue to question activity or finish.

### 3.4.4 Notification design

Notification will be shown as a pop up box, as seen in figure 3.5. During the quiz activity the notification will be shown to the participant. If the participant click the notification then the phone will be redirected to another application.

The notification should have the following properties:

- *shift* : this property is to decide on which shitft the notification will be shown, it will be compared to the *shift* properties of the study object.
- *phase* : this properties is to decide on which activity (question, answer and fill answer) the notification will be shown.
- *app* : what application the notification will open, it also need to have a value of the user or url of the application.
- *timeToshow* : how millisecond the framework should wait before the notification will be shown
- *notification text* what is the message text inside the notification box

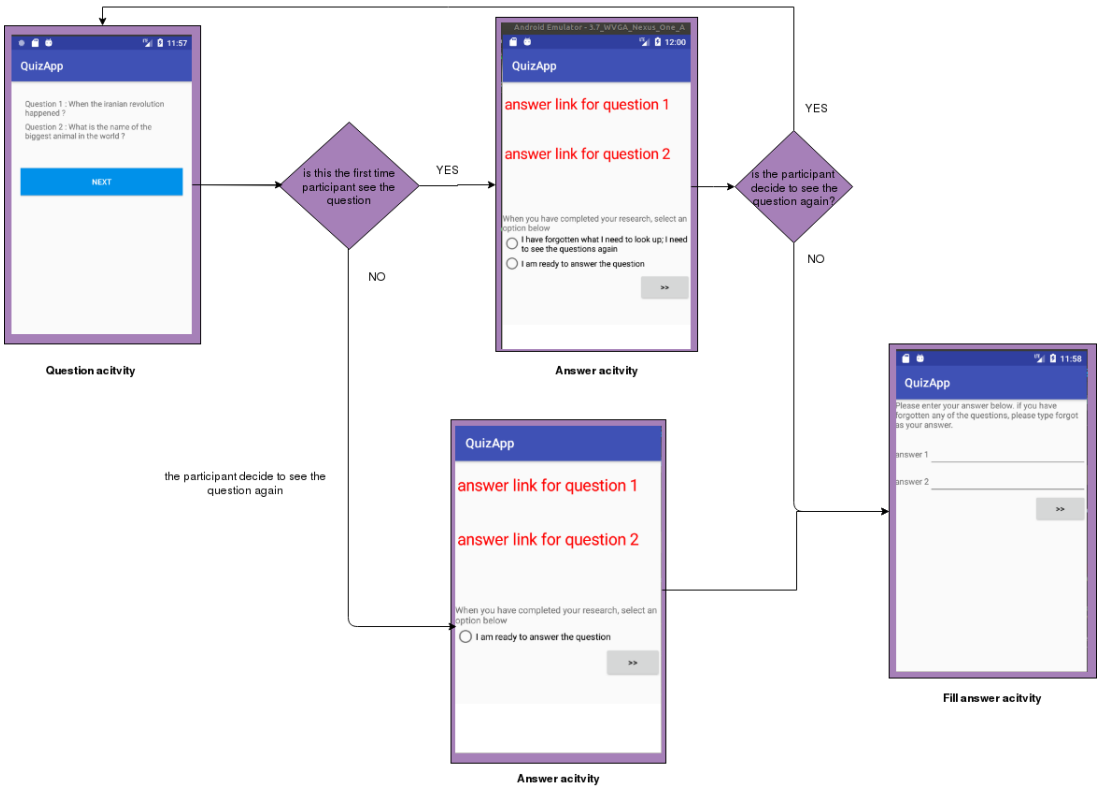


Figure 3.3: Quiz activity flow

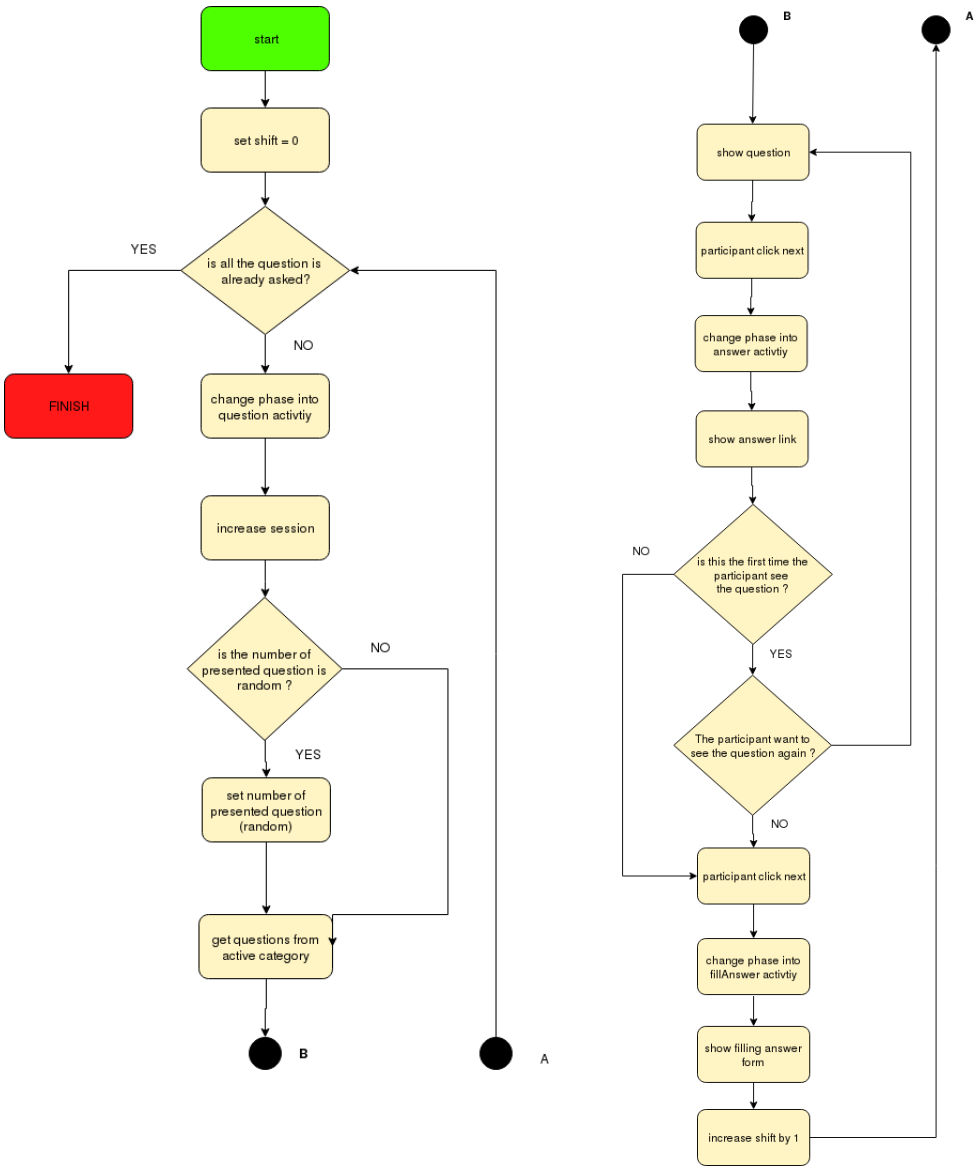


Figure 3.4: Quiz flowchart

All of these parameters above can be set by the researcher from the input file.

make this as a table

Figure 3.5 show the flow of the notification. The notification will be shown during the experiment process. If the user click the notification then the experiment application will be minimized and the android phone will be directed to another application. Then, the user can click the application icon to get back to the experiment application.

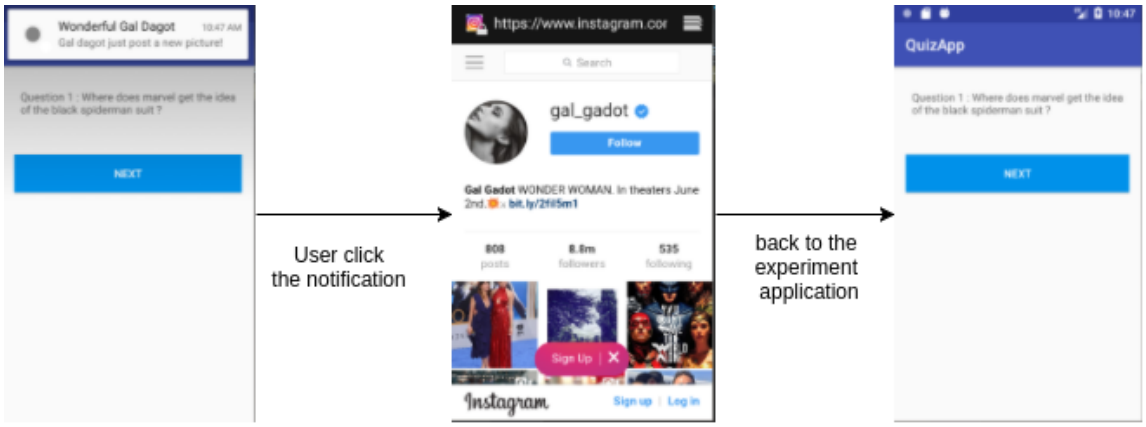


Figure 3.5: Notification flow

### 3.4.5 tracked variable

During the experiment the application should track variables. The following table consist of all the variables that the application track during the experiment. some of the variable has lb in front of their name, this mean that variable is tracked during the lookback process. A process when the participant look the question again for the second time.

| No | Question | Answer options |
|---|---|---|
| 1 | Often people go into a room to do something. Though they know they intended to do something,they lose track of what they wanted to do. This same sort of thing can happen when using a smart phone, as well. During the study, you may have clicked on a link, gone to the website, and then forgot what you intended to look up. Did that happen to you at all during this study? | Yes, No |
| 2 | During this study, did you ever look up an answer, then forget the answer before you were able to type it in? | Yes, No |
| 3 | During the course of this study, how many cell phone notifications did you receive ?' | 0,1,2,3 or more |
| 4 | How many notifications did you decide to click ? | 0,1,2,3 or more |
| 5 | As you were looking up information, did you ever follow a link you didńt need to follow, just out of interest ? | Yes,No |
| 6 | During the study, did you read about or learn any new facts that were not answers to questions we asked ? | Yes, No |
| 7 | How old are you ? | - |
| 8 | What is your gender ? | - |
| 9 | What country are you come from ? | - |
| 10 | Is English your native language | Yes,No |
| 11 | How many years have you spoken English? If you are a native speaker please leave blank. | - |
| 11 | What kind of phone do you nor- | non-smartphone, iphone, android, other |

| Requirement List | | |
|---|---|---|
| No | Name | Text |
| 1 | Upload Input | Researcher upload an .json input file to the application |
| 2 | Download Output | Researcher download a .json output file of the experiment |
| 3 | Set experiment properties | Researcher able to set extra experiment's properties apart from input file.<br>• study name<br>• participant Id<br>• researcher Id<br>• which experiment to conduct |
| 4 | insert multiple category | Researcher insert multiple categories |
| 5 | insert questions | On each category researcher insert multiple questions |
| 6 | set number of presented question | Researcher set how many question will be presented on each quiz phase |
| 7 | set presented question behavior | Researcher set whether the number of presented will be random each phase |
| 8 | insert post question | Researcher set whether the number of presented will be random each phase |
| 9 | insert notification | Researcher set notification with properties<br>• phase : which ordinal of the question will it be shown<br>• activity : what activity will it be shown<br>• how many millisecond it takes to wait before shown to the user |
| 9 | see the questions | the participant can see the questions |
| 10 | show answer link and answer page | the participant can clicked the answer links which will be |
| 11 | fill the answer | the participant can answer by filling the answer box |
| 12 | show notification | the application can show and pop up notification |
| 13 | Track variables | The application can tracked/logged variables |

Table 3.4: List of requirements

| Input | | | |
|---|---|---|---|
| No | Name | Type | Description |
| 1 | Study.PreText | String | a html string that will be shown at first on the experiment |
| 2 | Study.PostText | String | a html string that will be shown after the pretext |
| 3 | Study.NumExp | Integer | Researcher able to set extra experiment's properties apart from input file. |
| 4 | Study.Name | String | The name of the study |
| 4 | Study.Id | String | Researcher insert multiple categories (Optional) |
| 5 | Experiment.Name | String | The name of the experiment |
| 6 | Experiment.NumQuestion | Integer | Researcher set how many question will be presented on each quiz phase |
| 7 | Experiment.MaxPresentedQuestion | Integer | Researcher set whether the number of presented will be change randomly in each phase |
| 8 | Experiment.RandomPresentedQuestion | Boolean | Researcher set whether the number of presented will be random each phase |
| 9 | Category.Id | String | Id of the category (Optional) |
| 10 | Category.Name | String | The name of the category |
| 11 | Category.TotalQuestion | Integer | The total size of the question on this category |
| 12 | Category.QuestionOrder | String | the order of how the question will be pulled from the list of question. "LINEAR" it will be pulled based on the input order, "RANDOM" it will be pulled randomly |
| 13 | Category.Question.Id | String | the unique Id of the question |
| 14 | Category.Question.Text | String | The question text |
| 15 | Category.Question.linkAnswer | String | the http/https link of answer |
| 16 | Category.Question.Answer | String | the answer of the question |
| 17 | Notification.App | String | What application the phone will open if the participant click the notification |
| 18 | Notification.shift | Int | The number of phase when the notification should be shown. This will be explained more on the Notification section |
| 19 | Notification.Phase | String | The activies name when the notification should be appeared |
| 20 | Notification.TimeToShow | Integer | how millisecond the application should |

| No | Properties name | Description |
|---|---|---|
| 1 | *Shift* | This variable is used to count how many question-answer had been done. |
| 2 | *Phase* | This variable has a value on what activity is currently active on the application |
| 4 | *Active Category* | The active category that will be choose by the participant during the experiment. This category will contain list of questions. |
| 5 | *Active Experiment* | This variable will contains the properties of the current experiment, this properties is set by the researcher from the application |
| 6 | *Number of presented question* | this variable value shows how many question are asked at one *shift* of question-answer. *Active Question* : the current question and it's answer link that is presented to the participant. This active question is picked from the question list inside the active category. the number of active question is based on *number of presented question* variable. |

Table 3.7: List of application entity for the quiz activity

| Tracked variables list | | | |
|---|---|---|---|
| No | Variable's name | Type | Description |
| 1 | TTLQ | Long | total time (in millisecond) when the participant to see the question then click next button |
| 2 | lb_TTLQ | Long | similar with TTLQ and the participant decide to look the question again |
| 3 | LookBack | Boolean | True if the participant decide to look at the question again, false otherwise. |
| 4 | TTLB | Long | total time (in millisecond) when the participant see the answer links and decide to look the question again or answer the question (next button) |
| 5 | lb_TTLB | Long | Similar with TTLB and the participant decide to look at the question again.Then, answer links are shown again to the participant |
| 6 | visited_links | List of String | The list of links clicked/visited by the participant after clicking the answer links |
| 7 | time_visited_links | List of Long | List of the total time (in millisecond) the participant stay on a page after clicking link |
| 8 | lb_visited_links | List of String | similar with visited_links but the participant have decided to see the question again then return to the answer links window |
| 9 | lb_time_visited_links | List of Long | Similar with time_visited_links but the participant decide to look at the question again then click the answer links |
| 10 | TTLA | Long | Total time (in millisecond) the participant see the fill answer window and then click next |
| 11 | TTLFA | Long | Total time (in milisecond) the participant write the answer on the text box |
| 12 | num_notif | Integer | how many notification is shown |

# Chapter 4

# Implementation

This chapter discussed the technical implementation of the system based on the design discussed before. Section 3.4.1 provide the technical information about the main entity as classes that will be used in this study, including its properties and methods. section 3.4.1 to 3.4.5 provides Information about the class activities of the application.Section XX provides an information about the notification. Lastly, section XX provides an information about the tracker.

put the chap number

Flask framework and python programming language is used to develop the web-server, and java programming language and Android SDK is used to develop android application.

## 4.1 Entities relationship

All the entities discussed on the design will be represented as a class which consist of properties / variables and methods. The relationship between classes can be seen on the class diagram on figure 4.1. The box represent the class and the upper part represent the properties name and it's type. The plus and minus sign before the properties name represent the scope of the variable, minus (-) means private and plus (+) means public. The bottom part consist of the class's methods and the type of it's output. Furthermore, The arrow represent the properties relation with it's class type whether it has one-to-one (1..1) or one-to-many(1..*) relationship. It also represent that a class extend other class (it has same properties and methods). The class will be explained much detailed on the section below.

this is useless, should be deleted becaus eit's already explained inside

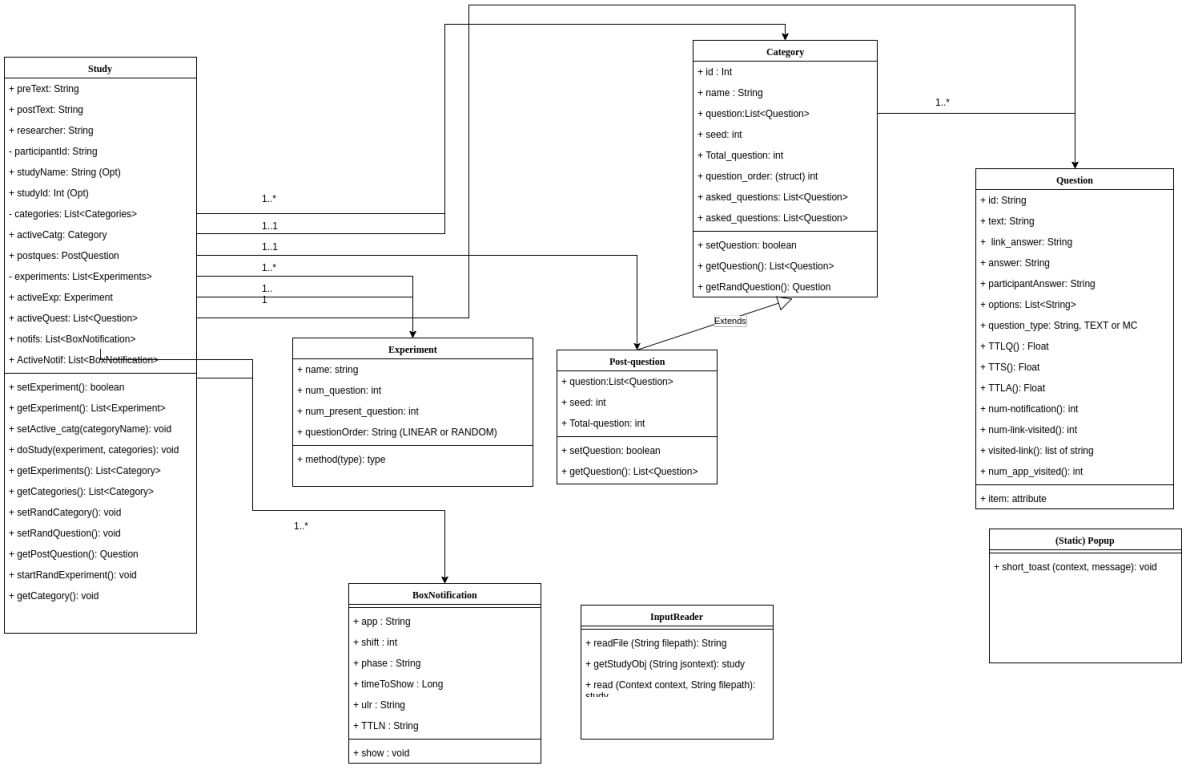


Figure 4.1: Class diagram

As seen in the UML class. the category class will have questions. Here Post question will has the same properties and methods as *category* class.

### 4.1.1 Study Class

Study class is the class that act as a main container for other entities of the application and it control the flow of the main function of the experiment. Some of its properties are defined from input file: *preText, postText, researcher, studyName, studyId*. As seen in the class diagram 4.1, the study class contains *experiments* variable which consist of list of experiment objects , *categories* variable which consist of list of category objects and *notifs* variable that consist of list of notification objects.

As explained in the design section, the experiment application need to initialize several variables before conducting the experiment. These variables define which experiment to conduct, which questions to show and which category to use. *ActiveExp* (active experiment) present which experiment is conducted. this variable set by the researcher on the application. And *ActiveQuest* (active question) present several questions being asked to the participant the value of *ActiveQuest* variable can consist of

several questions object which will be shown during the quiz experiment. The question inside *activeQuest* is picked from the list of question inside the active category (*activeCatg*) variable. The *activeCatg* variable is chosen by the participant during the experiment.

Moreover, the class also contains the active notification (*activeNotif*) this variable contains a list of notification that has been shown up to the participant. the notification come from the *notifications* variable that contains list of notifcations. the mechanism of the notification will be explained much further on the section **XX**.

mention which section is the notification

The study class contains methods that is used to control the main flow of the experiment. These methods will be explained on the next chapter.

which chapter explained the experiment

### 4.1.2 Experiment Class

The Experiment class contains the ***experiment properties*** that is used to define the behavior of the conducted experiment. The properties is explained more detailed in the table 4.2.

Every variables inside this class except *numPresentedQuestion* is determined from the input file. It is has one method **changeNumberPresentedQuestion()** which is called by the study class to change the *numPresentedQuestion* variable. The method will change the number presented question randomly from 1 to *maxPresentedQuestion* by using using the Random object (RandomGenerator) provided by java.

snapshot of the code

### 4.1.3 Category Class

Category class is used to carry the questions objects that has the same category. The two main part of the variable are *questions* and *askedQuestion*. *questions* variable simply is a list of questions, and the *askedQuestion* is list of question that has been asked to the participant.

The class has two main methods *getRandQuestion()* and *getQuestion()*. These methods will be called during the quiz activity to put question object into activeQuestion variable. These method aretwo different procedure to pick a question from list of question object inside *questions* variable. the former take the random randomly while

| No | Variable's name | Type | Description |
|---|---|---|---|
| 1 | name | string | The name of the experiment |
| 2 | numQuestion | Integer | The number of questions will be asked on the experiment |
| 3 | numPresentedQuestion | Integer | the number of question presented to the participant on the experiment every phase of question-answer |
| 4 | questionOrder | String | if the value is RANDOM, then the question is picked randomly from a list of question, if the value is LINEAR then the question will be picked based on the order of the input file |
| 5 | randomPresentedQuestion | Boolean | if it the value is true, then on each phase the num of presented question will changed randomly, explained more on changeNumberPresentedQuestion method |
| 6 | maxPresentedQuestion | Integer | this is the maximum number of presented question if the number of presented question is decided randomly |
| 7 | randomGenerator | Random | this is a random class that use to generate random nunmber, it is used inside changeNumberPresentedQuestion method |

Table 4.2: Experiment class variables

the latter will take the question based on the input order. After the question is selected, it will get deleted from the *questions* variable and thenput it into *askedQuestion* variable. The *getQuestion()* method get the latest question from the *questions* variable which is similar to the input order. On the other hand *getRandQuestion()* use the Random class the get the random index of the question.

| No | Variable's name | Type | Description |
|---|---|---|---|
| 1 | id | string | the id of the question |
| 2 | text | string | the text of the question |
| 3 | linkAnswer | string | the URL link to the answer page |
| 4 | answer | string | (optional) the answer of the question |
| 5 | participantAnswer | string | the answer of the participant during the quiz activity |
| 6 | questionType | String | The type of the question, the value can be "MC" or "TEXT". |
| 7 | representId | string | present id shows if two or more questions are presented together during quiz activity |
| 7 | options | list of String | List of the total time (in millisecond) the participant stay on a page after clicking link |

Table 4.4: Variable inside Question class

### 4.1.4 Question Class

The Question hold the questions, it's answer and the tracked variable that is tracked when the question is being asked. The variables is explained more on the table 4.6 , and the rest of the variable is explained on the tacked variable on the section XX.

what question to answer

The question can be two type MC or TEXT. MC means multiple choice, this question type will have multiple options on it's answer, and the participant can chose one of them. while TEXT means that the participant need to write the question.

where to get the traccked variabel

### 4.1.5 BoxNotification class

BoxNotification class present the Notification. BoxNotification is used as a class name because Notification is already defined class. The variable inside the class is explained on the table X.

The Notification can open an android application of twitter, facebook, instagram and web browser. The application should be installed to the phone, otherwise the notification will open the url of the application on the web browser. This app need to be specified on the *app* variable and the *url* variable. The *url* variable need to be filled with the user id of the twitter or instagram. Or it can be filled with http/https url to open web page. the class contains a show() method that will pop up the notification in the android phone. The the mechanism and flow of the notification will be explained on chapter XX

where is the notification flow

## 4.2 Application flow

In this section the flow of the application on the technical aspect will be discussed. The flow of the application can be seen on the figure **??**. The application is divided into four scope.

- Setting study : this part the researcher can set the properties of the experiment and choose to start the experiment.
- Experiment : this part when the participant conduct the experiment.
- Notification : this part when the notification can be shown up on the activities.
- PostQuestion : this part when the participant presented with post questions.

Need to write more about the bullshit here

Each class activity and method on the flow chart is explained further on the sub-section

### 4.2.1 Android Activity

The android application built upon multiple class activities. Each activities has an user Interface (UI) template. the UI consist of button, text, etc. And its corresponding class activity will decide what will appeared on the phone or what happen if a button is clicked . The event such us clicked or move to another application can be detected and linked to an *event listener* methods which will get called every time the event happened.

| No | Variable's name | Type | Description |
|---|---|---|---|
| 1 | app | string | The application that can be opened by the notification |
| 2 | shift | Integer | The number of questions will be asked on the experiment |
| 3 | phase | Integer | the number of question presented to the participant on the experiment every phase of question-answer |
| 4 | timeToShow | String | if the value is RANDOM, then the question is picked randomly from a list of question, if the value is LINEAR then the question will be picked based on the order of the input file |
| 5 | url | Boolean | if it the value is true, then on each phase the num of presented question will changed randomly, explained more on changeNumberPresentedQuestion method |
| 6 | titleText | Integer | this is the maximum number of presented question if the number of presented question is decided randomly |
| 7 | msgText | Random | this is a random class that use to generate random nunmber, it is used inside changeNumberPresentedQuestion method |
| 8 | presentedID | Random | this is a random class that use to generate random nunmber, it is used inside changeNumberPresentedQuestion method |

Table 4.6: Variable inside Question class

Figure 4.2 show how the android activity lifecycle. There are three main activities method used in this experiment application. *OnCreate()* is the first method to get called everytime the activity start. *OnPause()* will be called if the user of the phone move / redirected to another application. Lastly, *OnResume()* is called if the user open the application again after leave the application.

### 4.2.2 expInitialActivity

The UI layout of these activity can be seen in figure **??**. The participant can click a button to begin the experiment or to set the properties of the experiment.

On the OnCreate() event listener of the activity the InputReader.read() method is called. This method will read the input data and compiled it into Study object. The input data as string is compiled into json object by using gson library. Gson is a serialization / deserialization library that is used to convert string of object into json or another way around.

After reading the input file and compiled into The *Study* object. the object is need to be sent from an activity to another activity. *Intent* class of java is used to encapsulate the object and send to another activity. Because the *Study* object need to be encapsulated inside the Intent, java programming language require class of the object should implement the Serializable class.

snapshot of the intent code

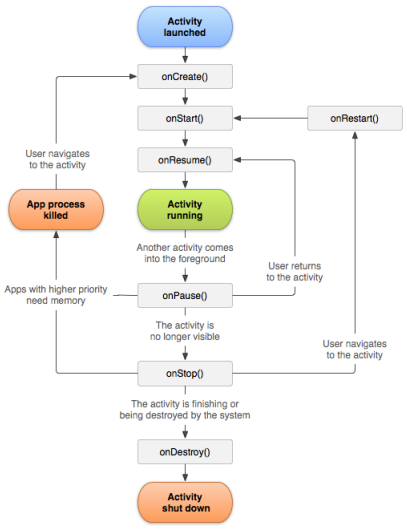


Figure 4.2: The cycle of activity

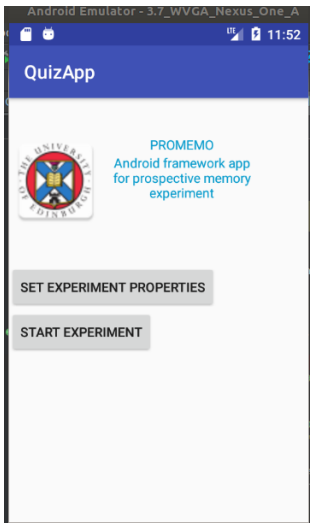


Figure 4.3: Caption

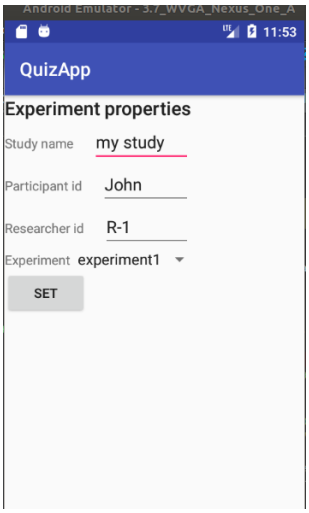


Figure 4.4: Caption

### 4.2.3 expSetPropActivity

This activity is used to set some parameter of the experiment. In this activity the researcher can pick which experiment to conduct and change the participant name. The purpose of this activity is to make it easier for the researcher to conduct multiple experiment and multiple participants without uploading the input file again. Figure XX shows the

Do I need a snapshot of code here?

### 4.2.4 IntroActivity

This activity is used to show the information about the experiment to the participant before starting the experiment. These information is stored inside the preText and postText string. the value of this variable will be converted into html and shown respectively. Figure X show example of the Consent information shown inside the application.

### 4.2.5 ChooseCategoryActivity

In this activity the participant choose which category he/she want to answer, as seen in the figure XX.

Also do I need to attach my code here?

Radio button represent the options of the category in the UI class. the selected

category name will then save in the selectedCategoryName variable inside study class as seen in the code XX.

### 4.2.6 QuestionActivity

In this activity, the question inside activeQuestion variables are shown to the participant. This activity will be called multiple time during the quiz experiment. the activity mainly call Study.runExperiment() method on the OnCreate() event. this method is the main method of the experiment, it use to start or continue the quiz experiment. this method is explained on the section below.

#### 4.2.6.1 Study.RunExperiment()

This method will be called everytime in the beginning of the Quiz activity. The main function of this methods are :

- Initialize the active experiment (*activeExp* variable) and active category (*activeCatg*) variable.
- Change the number of *presentedQuestion*, the variable is explained in the Experiment class.
- Set the active questions (*activeQuest*) from the questions in the category.

Figure 4.6 show the flowchart of the method. The First step is to initialize the experiment by checking the active experiment (activeExp) variable. if it's empty than it's mean that this is the first time the experiment run and it needs to initialize. Two main experiment properties which are active experiment (activeExp) and active category (activeCatg) are initialized by calling **initializeExperiment** method. The **initializeExperiment** method will initialize the experiment based on the *selectedExperimentName* and *selectedCategoryName*.

the **setRandomPresentedQuestion()** is called, this method will set the value of numPresentedQuestion which is the number of question presented on each phase of question-answer quiz. If the resercher set randomPresentedQuestion to true in the input file, then the value of number of presented variable will be random in the range of 1 to MaxPresentedQuestion. On the other hand if it's false, then the numPresentedQuestion will be constant.

Next, **isExperimentIsStillGoing()** is called. This method make sure if the experiment is still in on on going by checking that if the size of the *question* variable inside

the activeCatg is still larger or similar to *numPresentedQuestion*. so it's still possible to pick the question form the *questions* variable.

Lastly, the active question (*activeQuest*) is picked from the questions variable by calling **setActiveQuestion()**. this method will pick and put the question object from activeCatg into the *activeQuest* variable. the question object will be picked randmly if the researcher set questionOrder to random, otherwise the question will be pick linearly.

#### 4.2.6.2 AnswerActivity

In this activity the answer link are shown as a textview inside the UI layout. If the participant click the answer link then the clickListener inside the textview will open an answer window. An java class called *webview* is used as a browser of the answer page based on the URL of the answer link (question.url).

Two radio buttons are presented to the participant. These are the option whether the participant want to go back to see the question again or to continue to answer the question.

If the participant chose back then the application will go to question activity. on this transition the string is also capsulated inside the intent along with the study object. this string called "BACK" is used as a flag. This flag is send to question activity and send back to answer activity. this flag is used to know if the participant look at the question again. If the back flag is present on the answerActivity then the radio button to see the question again is hidden by setting it's visibility to GONE, as it seen on the snapshot of the code.

put the snapshot of the code

### 4.2.7 fillAnswerActivity

In this activity the participant should answer the question by writing the answer on the editText UI as seen on figure.

each editText is correspond with one question presented before, there can be multiple editText because there can be multipe question.

If the participant click next button then saveAnswer() method is called which will get the value of the editText and stored it on the participantAnswer on activeQuest.

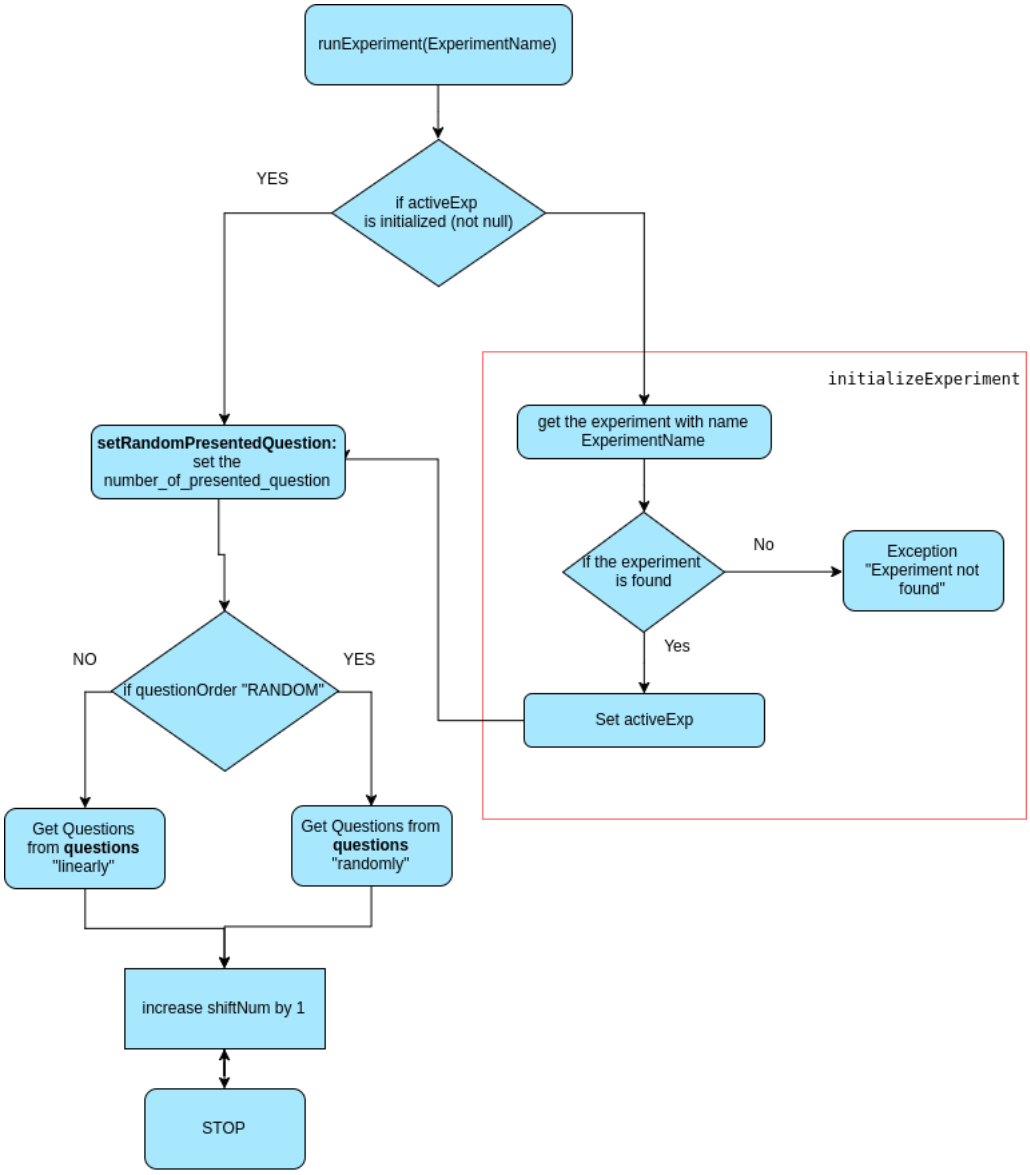


Figure 4.5: runExperiment flow chart

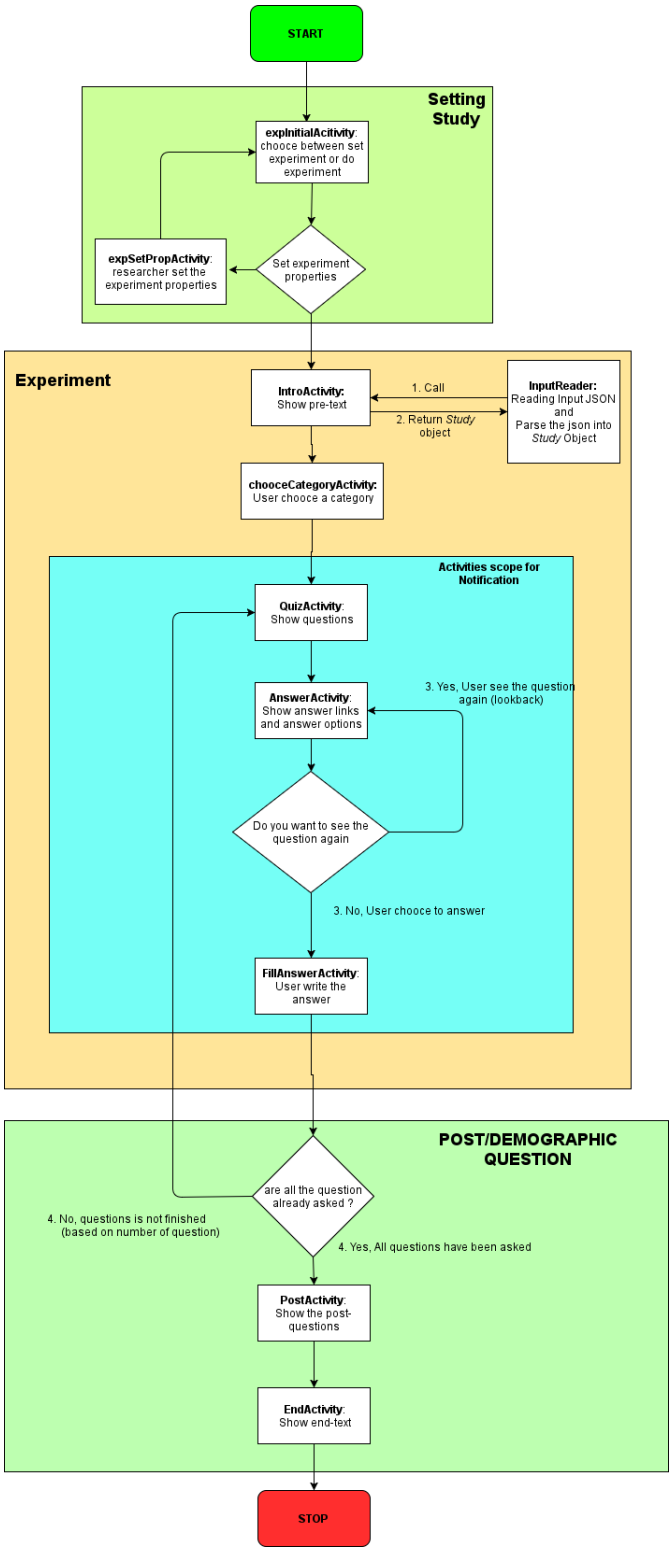


Figure 4.6: The flow of the application

## 4.3 Notification mechanism

Figure 4.7 shows the mechanism of notification. As seen on the flowchart, the Study.checkNotification() method is called inside the OnCreate event on the QuestionActivity, AnswerActivity and FillAnswerActivity. The method check on every notifications inside the *notifs* variable if the there is a notification that should be shown up based on phase and shift variable of the notifiaction. these variable is comapred with the variable passed by the method.

If there is a notification that need to be shown up then the notification object is added into the activeNotif and it is deleted from the notifs variable of the study object.

The notification need to wait for some millisecond before it can be shown up. The waiting time is defined in *timeToshow* variable inside the BoxNotification class. While the notification process wait, the main activity (Quiz) should keep working so another process need to be made a part from the main process. To accomplish it, TimerService class is used. This class will be spawn as another process and the process will be sleep for *timeToShow* millisecond then it will call the broadcastReceiver method which is defined inside each activity. This method will called the show method inside the notification.

The **Notifaction.show()** method is used to shown the notification to the front end of the android screen. This methode use NotificationCompact.Builder is used To build the notification layout. an Intent object is inserted inside the Builder object that contains what application to open. If the participant click the notification then the experiment application will be minimized. this event will call OnPause() methode on the current activity, and the android phone will open the intended application. The participant can go back to the current app by opening the application again and the OnResume() method of the current activity is called.

## 4.4 Tracker

The tracked variable are shown on the table 3.9. All of these variable are stored inside the Question class as it seen in the class diagram 4.1.

Some of the variables use to track the time in millisecond. To track the time the StopWatch class provided by java API is used. The StopWatch object is stored inside the study class because the StopWatch class is not serizable which means it can not be pass in the Intent object inside the Study class.

During the experiment the application will be minimized if the participant click the notification. During this event some Stopwatch object need to be pause. the OnPause() method of the curent activity is called when the application is minimized which is used to suspend the stopWatch and it will be resume again after OnResume().
this need to be explained

As it seen in the class diagram, each one of tracked variable has it's own StopWatch object for example StoWatchTTLQ will track the time for TTLQ variable. to stored the time tracked by the stopwatch the Study.log() method is called. this method pass two arguments, what variable to track and the stopwatch object used to track it. for instance log("TTLQ",stopWatchTTLQ) will track the TTLQ variable inside the activeQuest and use stopWatchTTLQ to get the tracked time. How each variable is tracked is explained further on the subsection below.

#### 4.4.0.1 TTLQ, lb_TTLQ and lookback

These variable is tracked inside the QuestionActivity. StopWatchTTLQ and stopWatchTTLQ_lb is used to track the timing of this variable. The stopwatch object start to count the time when OnCreate() method is called on the current activity, and the stopwatch will be stopped when the participant click next button. The Study.log() method will be called to track the variable when the next button is clicked. Inside the Study.log() method, it will call logTTLQ or logTTLQ_lb methods to stored the value of the tracked variable on each question object inside the activeQuest. the stopwatch variable will be passed on these method and the milisecond didapat bycalling StopWatch.getTime() method. The lookback variable will have true value if the participant chose to see the question again. These variable is set on the logTTLQ_lb() method.

### 4.4.1 TTLB and lb_TTLB

These variable are tracked inside the AnswerActivity. Similar with TTLQ, stopwatchTTLB and stopWatchTTLB_lb is used to track these variable. The stopwatch will start on Oncreate() method and it will be tracked when the participant click the next button.

### 4.4.2 visitedLinks, lb_visitedLinks, timeVisitedLinks and lb_timeVisitedLinks

These variable is tracked inside the AnswerActivity.

During the AnswerActivity if the participant click the answer link then the *webview* class of java is used to open the URL link. StopWatchLink then will be started, and it is used to track the time participant have spent on each web page inside the webview.

The mechanism of the tracking is shown in the figure 4.8. Firstly, the prevUrl variable is initialized, this variable stored the previous link the webview had opened. This webview class has an event listener called onPageFinished() which will called every time the web page has been finish loaded for example when the participant click the answer link or visit another link on the webpage. Every time the event listener onPageFinished() is called, or when the participant click the button then updateVisitedLinks() method is called. updateVisitedLinks() method stored the value of prevUrl (because it has been visited before) to visitedLinks variable and how long the participant spent on the web page to timeVisitedLinks.

check again

### 4.4.3 TTLA and TTLFA

These variable are tracked on the fillAnswerActivty. the TTLA simply tracked similar to TTLQ and TTLB, the variable tracked using StopWatchTTLA.

TTLFA is tracked differently because if there is more than one question then there will be multiple editText for the answer field.

On each editText element, the event listener called OnFocusChangeListener is attached to it. this event listener will be called if the is a change of focus on the UI layout of the activity, for example if the user click one editText then click another one. the event listener method can give an UI id of which editText was the user writing on before moving to another editText. the UI id will be stored inside the activeViewId variable.

should I attach some code. After getting the UI id a stopwatch corresponding on the editText should be obtained to track the time. To accomplish this stopWatchTTLFA is made as a hashmap where the key is the UI id, the id is made from the index of the question inside the activeQuest variable, and the value is the stopwatch correspond to the editText answer. so to access which Stopwach object correspond to particular editText the UI id from the event listener will be used to get the corresponding stopwatch object.

to track the time updateTTLFA() method is called this method will track the TTLFA time by accessing the stopwatchTTLFA hash map using the activeViewId variable.

jesus is hard to explain this

### 4.4.4 numNotif, numNotifClicked and TTLN

As explained on the section 4.3.8 during the QuestionActivity, AnswerActivity and fillAnswerActivity the checkNotifaction() will find the notification that should be shown up to the screen and it will called the inceraseNumNotif() method. This method will increase the numNotif variable of every question inside activeQuest.

If the notifaction is clicked then the inceraseNumNotifisClicked() method will be called inside the broadCastReceiver on the current activity class. This method will increase the number of numNotifClicked variable on every question object inside the activeQuest variable.

notifStopWatch object is used to track the TTLN time. After the participant click the notification than the application will be minimized and other application will be opened. As explained on the Android activity section (4.3.1), the OnPause() event handler will be called just before the application is minimized. On OnPause() method a Study.startLogNotif() method is called, this method will start the notifStopWatch object. Then after the participant got back to the experiment application the onResume() method is called. This method will call Study.stopLogNotif() method. This method will get the current notification and set the TTLN from the notifStopWatch. Kliegel and Martin (2003)

increase the font

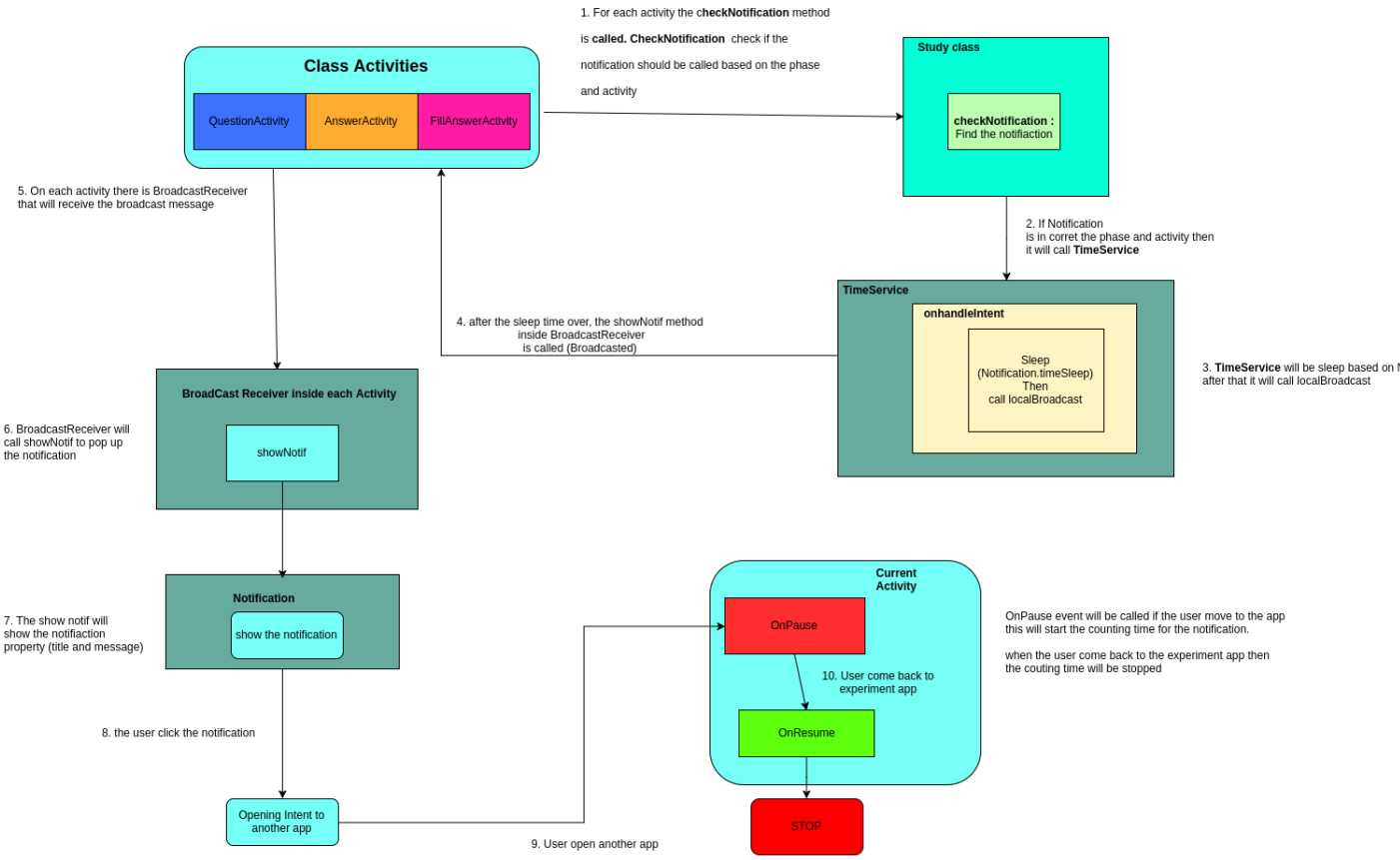


Figure 4.7: The flow of the notification

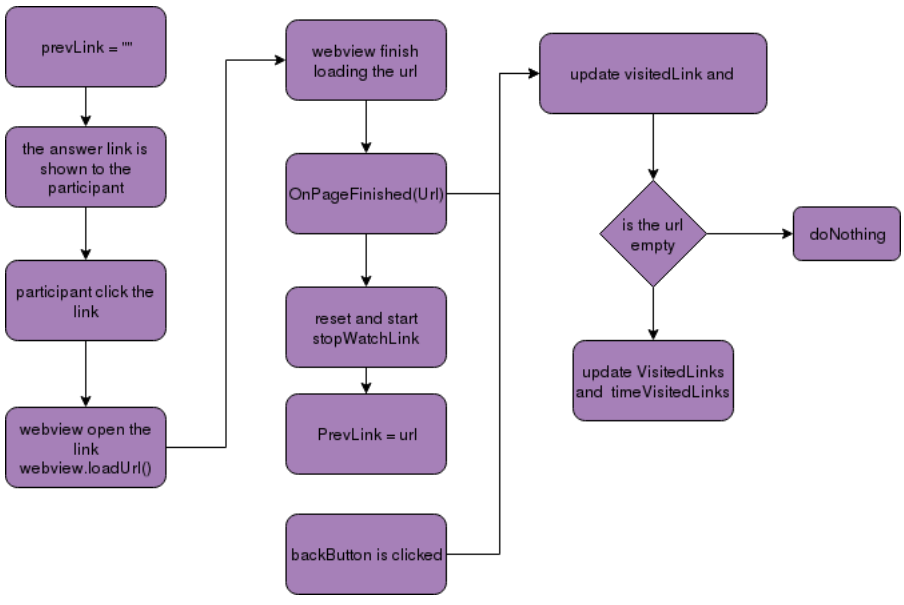


Figure 4.8: tracker mechanism for webview

q

# Chapter 5

# Experiment Result

## 5.1 Experiment output

show the result of the experiment data

is the participant experience prospective memory while doing the smartphone ? using general data. using all data look at the participant answer. and look at the real fact, that how many people experience lookback. This is shows that using while using a smartphone people experience the prospective memory error.

so we are trying to analyze two stuff here, one intention vs two intention

and how can the notification make people forget

and how moving from one side to another side make people harder these three study study 1 , average TTLFA, $\text{time}_{visited_answer, lookback}$

## 5.2 Data analysis

This is where I explain the result of the experiments

Table 5.1: My caption

| No | Study 1 | | | Study 2 |
|---|---|---|---|---|
| | Claim that they lost intention | Number of person forget the question | Total Person | Claim |
| 1 | 0 | 3 | 4 | 5 |

Figure 5.1: Participant answer on the prospective memory error on smartphone (study 2)

Table 5.2: My caption

| Question | Study 1 | | Study 2 | | Study 3 | |
|---|---|---|---|---|---|---|
| | Yes | No | Yes | No | Yes | No |
| Experience prospective memory error | 0 | 4 | 5 | 6 | 2 | 1 |
| Forget about the answer | 2 | 2 | 4 | 7 | 0 | 3 |

## 5.3 Hypothesis testing

## 5.4 Discussion

Table 5.3: My caption

| No | One Notification | | | Two Notification | | |
|---|---|---|---|---|---|---|
| | TTFA | time looking at answer page | lookback frequency | TTLFA | time looking at answer page | lookback frequency |
| 1 | 7292.92 | 69517.0 | 8 | 7304.87 | 76699.16 | 10 |

# Bibliography


Baddeley, A. and Wilkins, A. (1983). *Taking memory out of the laboratory.*, pages 1–17. Academic Press.

Brandimonte, M., Einstein, G., and McDaniel, M. (1996). *Prospective Memory: Theory and Applications*. L. Erlbaum.

Cockburn, J. (1995). Task interruption in prospective memory: A frontal lobe function? *Cortex*, 31(1):87–97.

Cockburn, J. and Smith, P. T. (1994). Anxiety and errors of prospective memory among elderly people. *British Journal of Psychology*, 85(2):273–282.

Cohen, G. and Conway, M. (2008). *Memory in the Real World*. Psychology Press.

Crovitz, H. F. and Daniel, W. F. (1984). Measurements ofevery day memory:toward the prevention of forgetting. *Bulletin of the Psychonomic Society*, 5:413–414.

Curiel, J. M. and Radvansky, G. A. (2002). Mental maps in memory retrieval and comprehension. *Memory*, 10(2):113–126.

Einstein, G., J. Holland, L., Mcdaniel, M., and Guynn, M. (1992). Age-related deficits in prospective memory: The influence of task complexity. 7:471–8.

Einstein, G. O. and Mcdaniel, M. A. (1990). Normal aging and prospective memory. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16(4):717–726.

Einstein, G. O. and McDaniel, M. A. (2005). Prospective memory. *Current Directions in Psychological Science*, 14(6):286–290.

Ellis, J. (1996). Prospective memory or the realization of delayed intentions: A conceptual framework for research. In Brandimonte, M., Einstein, G., and McDaniel, M., editors, *Prospective Memory: Theory and Applications*. L. Erlbaum.

Gauld, A. and Shotter, J. (1977). *Human Action and Its Psychological Investigation*. Routledge & Kegan Paul, Limited.

Groot, Y. C., WILSON, B. A., EVANS, J., and WATSON, P. (2002). Prospective memory functioning in people with and without brain injury. *Journal of the International Neuropsychological Society*, 8(05).

Guez, J. and Naveh-Benjamin, M. (2013). The asymmetrical effects of divided attention on encoding and retrieval processes: A different view based on an interference with the episodic register. *PLOS ONE*, 8(9):1–14.

Guynn, M. J., McDaniel, M. A., and Einstein, G. O. (1998). Prospective memory: When reminders fail. *Memory & Cognition*, 26(2):287–98.

Haenggi, D., Kintsch, W., and Gernsbacher, M. A. (1995). Spatial situation models and text comprehension. *Discourse Processes*, 19(2):173–199.

Heckhausen, H. and Beckmann, J. (1990). Intentional action and action slips. *Psychological Review*, 97(1):36–48.

Heckhausen, H. and Kuhl, J. (1985). From wishes to action: The dead ends and short cuts on the long way to action. In Frese, M. and Sabini, J., editors, *Goal Directed Behavior: The Concept of Action in Psychology*, pages 10–134. L. Erlbaum Associates.

Jacoby, L. L., Woloshyn, V., and Kelley, C. (1989). Becoming famous without being recognized: Unconscious influences of memory produced by dividing attention. *Journal of Experimental Psychology: General*, 118(2):115–125.

Kliegel, M. and Martin, M. (2003). Prospective memory research: why is it relevant. *International Jonral of psychology*, 38(4):193–194.

Kurby, C. A. and Zacks, J. M. (2008). Segmentation in the perception and memory of events. *Trends in Cognitive Sciences*, 12(2):72–79.

Kvavilashvili, L. and Ellis, J. (1996). Varieties of intention: Some distinction and classifications. In Brandimonte, M., Einstein, G., and McDaniel, M., editors, *Prospective Memory: Theory and Applications*, pages 23–51. L. Erlbaum.

McDaniel, M. and Einstein, G. (2007). *Prospective Memory: An Overview and Synthesis of an Emerging Field*. Cognitive psychology program. SAGE Publications.

McDaniel, M. A., Robinson-Riegler, B., and Einstein, G. O. (1998). Prospective remembering: Perceptually driven or conceptually driven processes? *Memory & Cognition*, 26(1):121–134.

Mcgann, D., Ellis, J. A., and Milne, A. (2002). Conceptual and perceptual processes in prospective remembering: Differential influence of attentional resources. *Memory & Cognition*, 30(7):1021–1032.

Radvansky, G. A. (2012). Across the event horizon. *Current Directions in Psychological Science*, 21(4):269–272.

Radvansky, G. A. and Copeland, D. E. (2006). Walking through doorways causes forgetting: Situation models and experienced space. *Memory & Cognition*, 34(5):1150–1156.

Radvansky, G. A., Copeland, D. E., and Zwaan, R. A. (2003). Brief report: Aging and functional spatial relations in comprehension and memory. *Psychology and Aging*, 18(1):161–165.

Radvansky, G. A., Krawietz, S. A., and Tamplin, A. K. (2011). Walking through doorways causes forgetting: Further explorations. *The Quarterly Journal of Experimental Psychology*, 64(8):1632–1645.

Radvansky, G. A., Tamplin, A. K., and Krawietz, S. A. (2010). Walking through doorways causes forgetting: Environmental integration. *Psychonomic Bulletin & Review*, 17(6):900–904.

Reason, J. T. (1979). Actions not as planned: The price of automatization. In Underwood, G. and Stevens, R., editors, *Aspects of Consciousness*, pages 1–67. Academic Press.

Reason, J. T. (1985). Lapses of attention on everyday life. In Parasuraman, I. R. and Davies, D. R., editors, *Varieties of attention*, pages 10–134. New York Academic Press.

Scullin, M. K., Bugg, J. M., and McDaniel, M. A. (2012). Whoops, i did it again: Commission errors in prospective memory. *Psychology and Aging*, 27(1):46–53.

Searle, J. (1983). *Intentionality: An Essay in the Philosophy of Mind*. Cambridge paperback library. Cambridge University Press.

Shorrock, S. T. (2005). Errors of memory in air traffic control. *Safety Science*, 43(8):571–588.

Swallow, K. M., Zacks, J. M., and Abrams, R. A. (2009). Event boundaries in perception affect memory encoding and updating. *Journal of Experimental Psychology: General*, 138(2):236–257.

Wilkins, A. and Baddeley, A. (1978). *Remembering to recall in everyday life: an approach to absentmindedness.*, page 247. Academic Press.

Winograd, E. (1991). Memory in the real world. gillian cohen. lawrence erlbaum associates, hove and london, 1989. no. of pages: 247. ISBN 0-86377101-7 (paperback). price: $19.95. *Applied Cognitive Psychology*, 5(5):247.