Summary Report on
Paper No.:-20

Usability implications in software architecture: The case study of a mobile app

Submitted to:
Dr. Tanmaya Mahapatra
BITS pilani
(Department of Computer Science)

Submitted by:
Group no. 2
Manish Mahajan (2022H1120275P)
Anand Kumar (2022H1120277P)
Srijon Mallick (2022H1120279P)
Ramananda Samantaray (2022H1120266P)

Introduction

Usability is a highly desired but often ignored software quality. Effective user interfaces tend to increase learnability, effectiveness and user satisfaction. But usability is often neglected in the early stages of software development and is thus frequently not addressed in a system's architectural design. Furthermore, because usability is often neglected or deferred, it is difficult to estimate the effort specifically spent on adding or improving usability mechanisms in the software architecture and in code. In this work, we present a case study where we analyze the impact of introducing a variety of usability mechanisms to a mobile application and we report on the architectural changes that must be made in the software architecture to accommodate them. We also report on the additional code required to implement these usability mechanisms and we investigate the users' satisfaction of combining and using several usability mechanisms in a mobile application.

Technical details

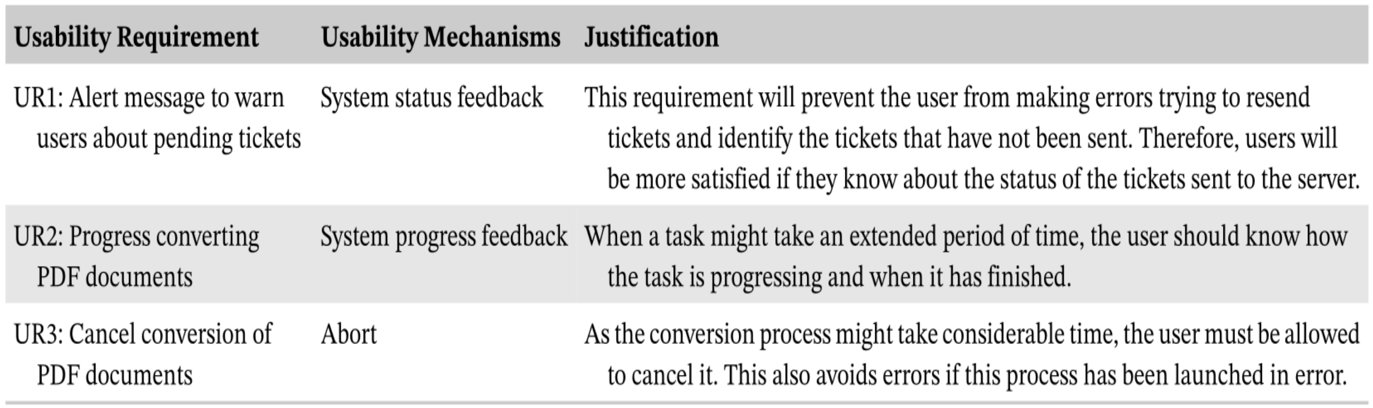
Case study experimental setup:-

1) **Mobile application:-** M-ticket application
   The target system consists of a modern Smartphone Android application which includes a standalone server system for managing the traffic tickets (M-Ticket). The handheld device is used by local police to capture traffic infractions and manage digital tickets which are sent to a Web application using a mobile phone. The application allows the policemen to capture an image and the location of the infraction via global positioning system (GPS) and introduce the data of the car and the infraction. Once the Web

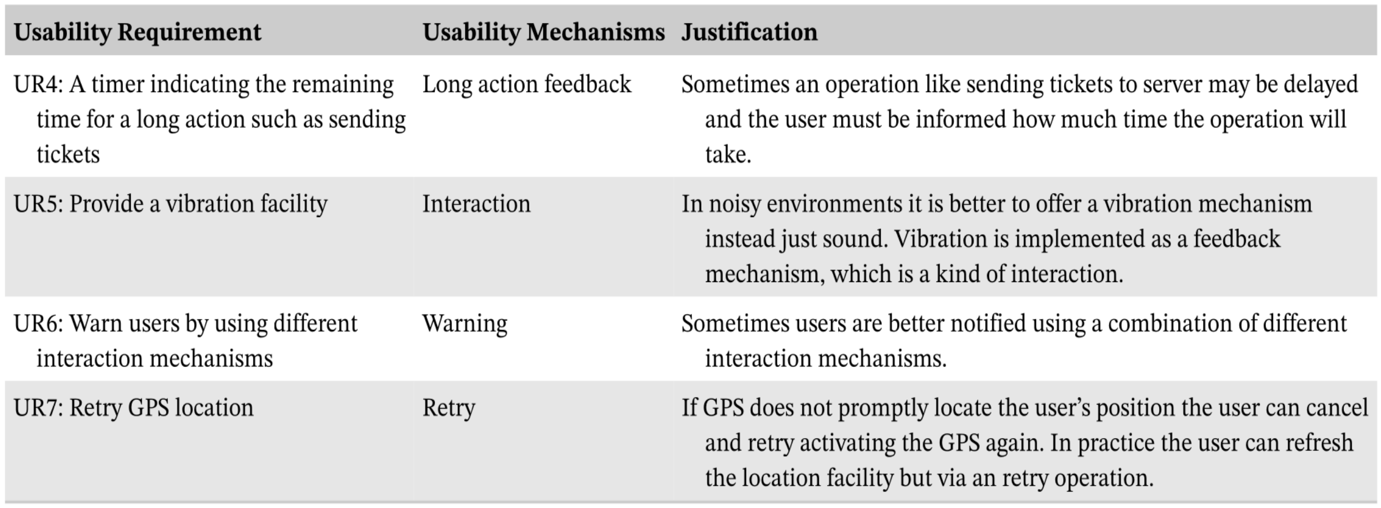
application receives an infraction in the police station, the data with the picture of the infraction and the map with the location are stored in a database server. In addition, an electronic ticket in PDF format is generated, which can be sent out to the owner of the car or printed out.

2) **No. of users:-** 25

7 usability mechanisms:-



Usability Requirement	Usability Mechanisms	Justification
UR1: Alert message to warn users about pending tickets	System status feedback	This requirement will prevent the user from making errors trying to resend tickets and identify the tickets that have not been sent. Therefore, users will be more satisfied if they know about the status of the tickets sent to the server.
UR2: Progress converting PDF documents	System progress feedback	When a task might take an extended period of time, the user should know how the task is progressing and when it has finished.
UR3: Cancel conversion of PDF documents	Abort	As the conversion process might take considerable time, the user must be allowed to cancel it. This also avoids errors if this process has been launched in error.

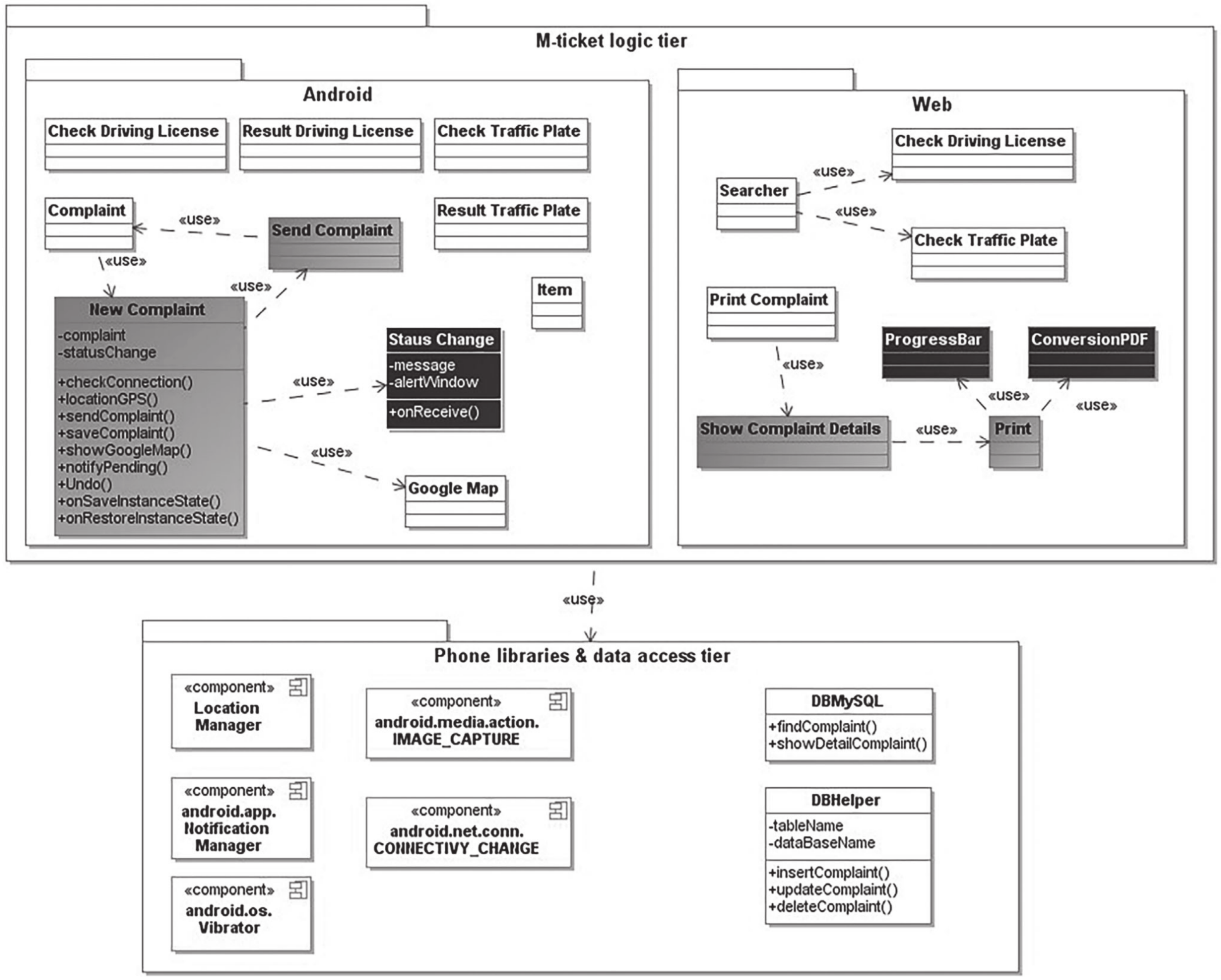


Usability Requirement	Usability Mechanisms	Justification
UR4: A timer indicating the remaining time for a long action such as sending tickets	Long action feedback	Sometimes an operation like sending tickets to server may be delayed and the user must be informed how much time the operation will take.
UR5: Provide a vibration facility	Interaction	In noisy environments it is better to offer a vibration mechanism instead just sound. Vibration is implemented as a feedback mechanism, which is a kind of interaction.
UR6: Warn users by using different interaction mechanisms	Warning	Sometimes users are better notified using a combination of different interaction mechanisms.
UR7: Retry GPS location	Retry	If GPS does not promptly locate the user's position the user can cancel and retry activating the GPS again. In practice the user can refresh the location facility but via an retry operation.

Objectives:-

This paper attempts to answer the following questions.
Q1:- What is the **impact to the architecture** when combining different usability mechanisms?
Q2:- What is the **additional coding effort** in implementing usability mechanisms?
Q3:- What is the **perception of users** using a combination of usability mechanisms in a mobile app?

Impact on architecture:-



Newly added classes and libraries are highlighted in the above layered architecture diagram.

Impact on additional coding effort:-

Changes in code due to status feedback mechanism:-

Classes	**LOC(before)**	**LOC of new code (after)**	**Total LOC(after)**
sendComplaint.java	354	15	369
NewComplaint.java	636	108	744

Changes in code due to progress feedback and abort mechanism:-

Classes	LOC(before)	LOC of new code (after)	Total LOC(after)
showComplaintDetails.php	71	9	80
print.php	353	74	427
progressbar.php	-	114	114
conversionPdf.php	-	14	14

Changes in code due to warning, retry, long action feedback and interaction mechanism:-

Classes	LOC(before)	LOC of new code (after)	Total LOC(after)
statusChange.java	-	57	57
NewComplaint.java	744	69	813

All in all, the modeling effort was not big as we did not need to change the architectural style chosen but only added some classes, methods and relationships supporting the usability mechanisms. These architectural changes are supported by the low extra coding effort as shown in the numbers discussed in Tables 4 and 5, which demonstrates that these changes were indeed not large.

<u>User's satisfaction on usability mechanisms:-</u>

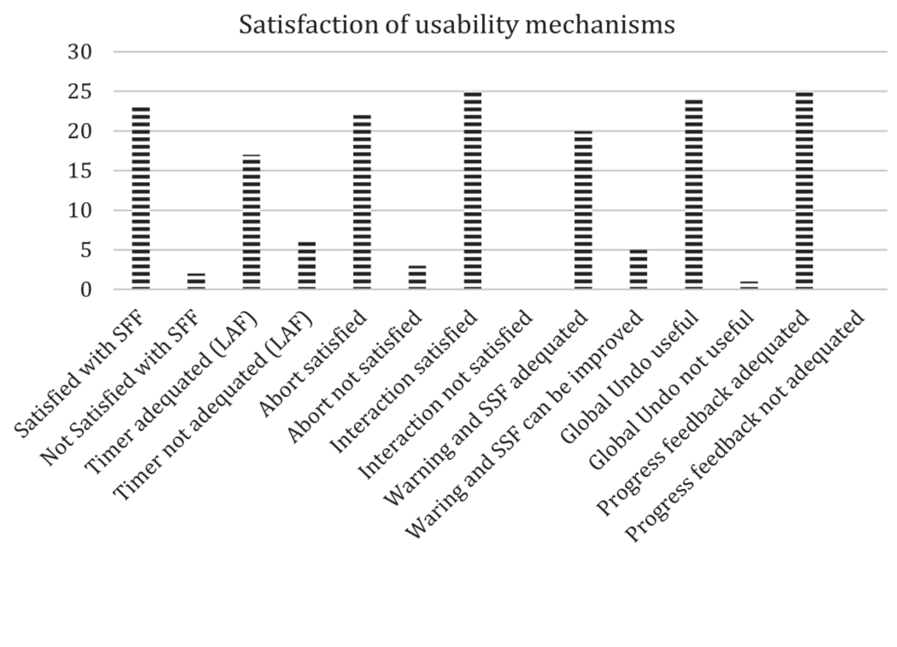


FIGURE 8 Summary of the perception and degree of satisfaction of the number of users about the seven usability mechanisms implemented and evaluated

Findings & limitations

The main findings we observed derived from the results and usability evaluation are as follows:-

1) It notes that while abort and retry mechanisms may be difficult to implement in complex applications with multiple state changes, they work well in simpler applications.
2) Users found the combination of vibration, sound, and visual effects to be useful.
3) The testing effort of some of the usability mechanisms was a bit higher.
4) On average, around two new classes and four new relationships per usability mechanism are needed.
5) it is difficult to estimate what would be the impact in the architectural responsibilities using a different architectural pattern such as MVC or pipe and filter.
6) Another interesting issue is whether the order of the usability mechanisms may impact the resulting architecture and the effort to implement it. We cannot provide additional insight about this problem as we did not perform attempt using a different order of implementation

Conclusion

There is a lack of previous research on incorporating usability mechanisms into software systems, particularly mobile apps, and the associated effort required for their implementation. The authors state that they have provided significant data and estimations regarding the impact of incorporating usability mechanisms on software architecture and the additional design and implementation effort required for each mechanism. This information can be useful for developers looking to build similar applications.

In addition, the usability tests, combined with contextual interviews, gave us precise feedback about the combination of the different mechanisms and whether the users felt that this combination was right or wrong or in which cases an improvement to a particular mechanism was needed or if too many mechanisms were used. Our results provided interesting feedback for software engineers to know if we need to combine mechanisms in a different way or those cases where a small modification (eg, increasing status messages) provides significant end-user benefit.

Future scope

Finally, for future work we plan to improve the mechanisms based on the feedback received and replicate them in similar apps to analyze the architectural changes needed and to estimate how the coding effort varies across different apps and architectural patterns.