

AutoInfo

Team: Ahmed, Nurlybek, Ruslan.

A dark blue diagonal gradient bar that starts from the bottom left corner and extends towards the top right corner, covering the lower half of the slide.

Idea of the application:

A- Awareness of automotive craftsmanship.
Simplicity of organizing & accessing details of vehicles based on make and model of car.

B- Users will be able to determine characteristics and specifications of vehicles by choosing a make and model.

C- Users are also able to create their account and interact with each other using the chat functionality.

D- Users are also able to find the car make locations based on their choice of the car make.

E- Users are guiding by Tutorial for more details of the use of the application.

Software Engineering Practices

The Agile practices that our team has been implemented are listed below, most of the principles we follow are taken from iterative agile methodologies called “Lean”, “Scrum” and “XP”

1) Working as a self-organizing team and encouraging role interchangeability

The roles in our team were assigned by the group members and tasks of each members may be exchanged between members if the task is not clear to the particular member or if team member is willing to trade the task. The results of implementing certain function or design feature would be written in Jira and could also be found in the github commits and comments

2)Continuous Integration

This practice was used by the group when we face difficulties to make the certain functionality work, therefore we were continuously integrating the alternative methods of implementing this functionality whenever break presented itself until functionality is up and running.

3)Rules of simplicity

In the iteration of the project our team wasl trying to populate the app with data from the API, however the simplicity in the source code is our first pryority, therefore we set rules that needs to be followed:

- the code is verified by automated tests, and all such tests pass
- the code contains no duplication
- the code expresses separately each distinct idea or responsibility

4)Pair Programming

During the development we were meeting regularly to work on the features of our app and started to implement pair programming such that one functionality was written by one team member while another was reviewing each line of code as it was typed.

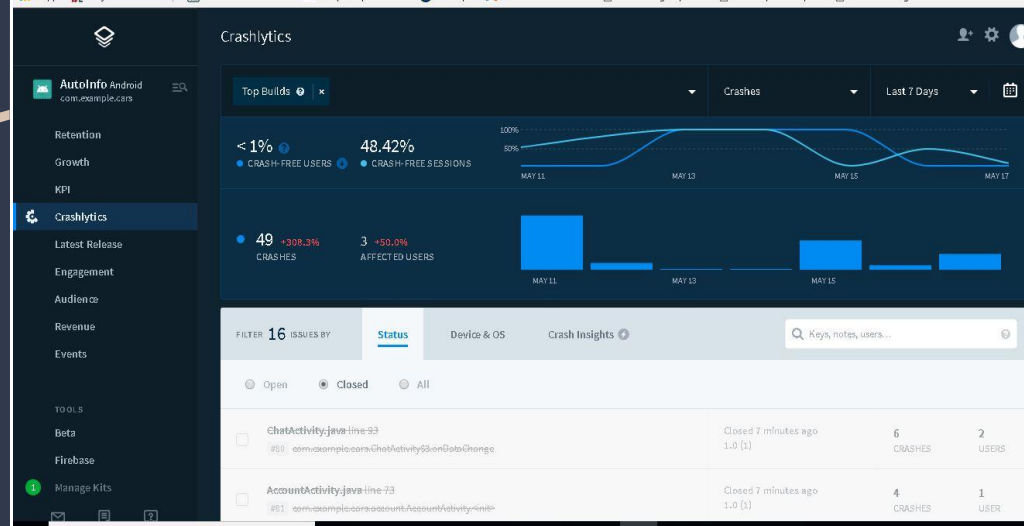
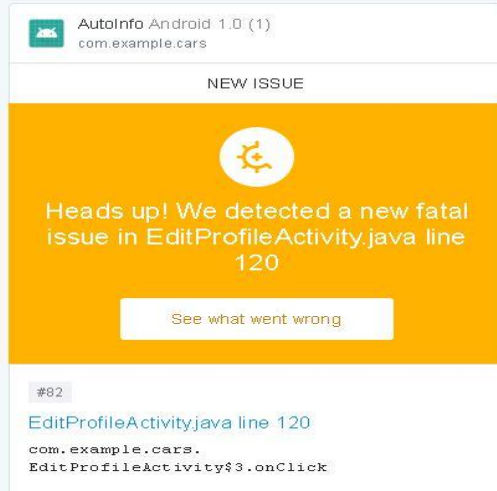
5)Heartbeat Retrospective

The team met regularly, adhering to the rhythm of the current iteration, to explicitly reflect on the most significant attempts that have occurred since the previous meeting and take decisions aiming at improvement

Quality assurance

1) Utilize Already-made Test Frameworks

Our team is implementing tools for the testing that were specifically developed for the android platform. The system we were using is called “Fabric”, Fabric lets developers run tests with modular kits that can be mixed and matched to build the glitch free application.



2) Break free from the classical roles and responsibilities

As a development team we are flexible to interchange the roles that we were assigned initially to identify potential flows before anyone spends time coding. In other words, any of the team member can be more familiar with the implementation of the certain functionality than others and can teach others how to implement it the best way possible. Concerning the design part, we usually ask each other if certain UI element fits the overall design properly or not.

3) Adopt 'Test Early Test Often' Principle

Our team was testing early and often, which helped us to find bug early and confirm continuous deliverable

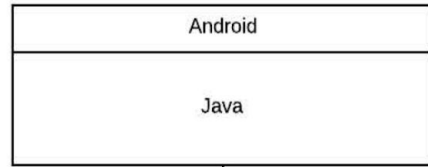
Revision control:

Our group was working with GitHub revision control system. The link to the github we have created for the project is: <https://github.com/ruslanski/AutoInfo>

Policies that we are utilizing in regards to revision control system:

- 1) Only push changes to the global master branch
- 2) Only push if the feature you added is working
- 3) Whenever you push, you should pull first to avoid merge conflicts
- 4) Whenever new activity is created you should put it in the folder designed for this type of the activity
- 5) Every activity layout XML file should be stored in the app/res/layout in the revision control system

Logical View



Business Layer

