Critical Analysis of Software Architecture version 1.0

HomePairs
Computer Science Department
California Polytechnic State University
San Luis Obispo, CA USA

November 12, 2019

Contents

Credits

Name	**Date**	**Role**
Luke Reckard	November 12, 2019	Co-author of Critical Analysis of Software Architecture
Joshua Boe	November 12, 2019	Co-author of Critical Analysis of Software Architecture

1 Technical Risks

1.1 Section 3.2

The deployment diagram specified displays all of the architecture that will be used for the HomePairs application. However, in the paragraph beneath it, there seems to be a lot of uncertainties that come with using the Google Cloud Application for a web application and for hosting a database. These should be clarified through more analysis of different databases and cloud-based servers.

1.2 Section 3.3

1.2.1 Database Schema

I am a little confused on how the Chat database table is going to work, although I understand it is a stretch feature. Maybe consider having a message GroupId and a MessageId as a composite primary key.

1.2.2 Class Diagram

The user roles for the class diagram are too vague and not specified for how they will be handled for each application or each account. It is also unknown whether the HomePairs application, the Roopairs API, or the OS will handle the user permissions. This should be clarified not just in the class diagram but in other places as well.

Both the PropertyManager and Property classes have a list of Tenants. This could be a code smell regarding data synchronicity if one list of Tenants is mutated but not the other. Since PropertyManagers already have a list of Propertys, maybe consider removing the list of Tenants from the PropertyManager class and then referencing Tenants from the list of Tenants within each Property.

2 Information Risks

2.1 Section 3.3

The diagrams seem to cover a lot of the flow through the application such as data and messages but do not cover the interactions between different objects. These gaps could be filled with sequence diagrams that would discuss how different objects interact with each other.

2.1.1 Class Diagram

The class diagram seems too simple for the scope of the project. Users were somewhat fleshed out but there were only two classes Property and Appliance. Although systems should be simple, the classes described seemed too simple and miss out on other functionality of the program, such as requests and invoices.

3 Economic Risks

3.1 Section 3.2

The paragraph for components specifies that Google Cloud Platform may not fit the goals of the customer in terms of a free tier for the database and other services. This is an economic risk as the spending should be clarified before deciding to use a framework, as the costs may be too expensive for the scope of the project beyond a free tier.

4 Managerial Risks

One of the project requirements of Roopairs includes having some kind of data analysis component that provides some kind of insight into either Jobs or Equipment or both. However, the Software Architecture v1 document does not currently consider this feature.

5 Other Critiques

5.1 Section 3.2

The deployment diagram should not contain development tools such as Android Studio.

5.2 Section 3.3

5.2.1 Class Diagram

There is a “Diagram.png” typo underneath the class diagram.

5.3 Section 4

These issues seem more like informational risks and should be addressed with the customer and analyzed more.

5.3.1 Glossary

Some of the terms in the glossary seem superfluous (’chat’ and ’notification’).