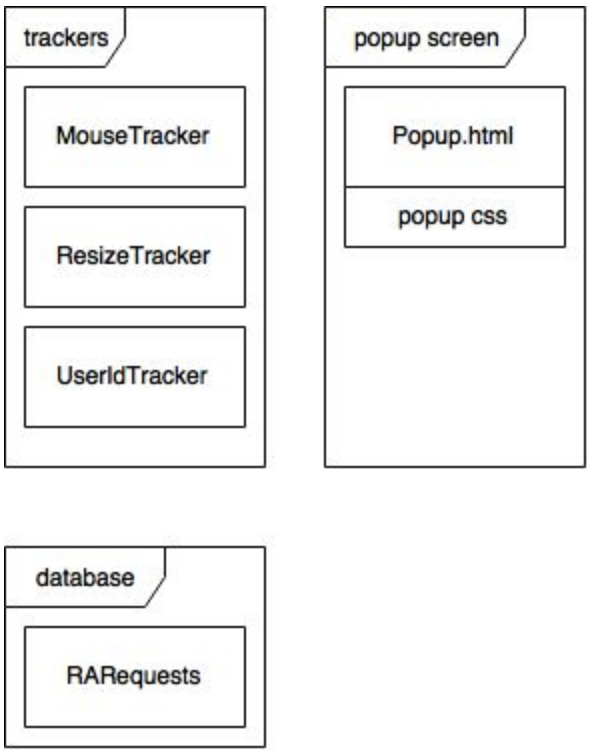
[This is a draft version]

Introduction

1.1 Design goals



Software architecture views

Subsystem decomposition (sub-systems and dependencies between them)

We expect the application to consist of a few components:

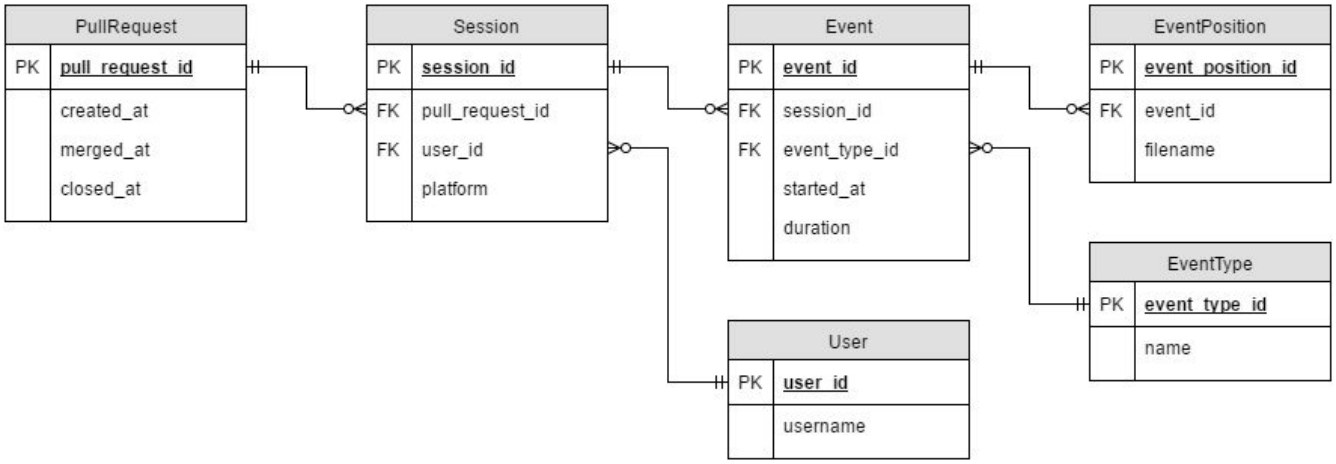
- **Trackers** keep track of the events we want data on, for example mouse movement, window resize, but also current user. Should rely on having a connector.
- **Connectors** provide a layer between the tracker and the AJAX calls towards the (database) server.
- The **user interface** is the component that the user interacts with, depends on having trackers to enable & disable.
- A possible extra subsystem is the **analytics component** which would provide a service to the user interface (and possibly other components) by providing information about - for example - a certain pull request and simple statistics. Requires a connector to fetch information from a central server or a connector to a local data source.

Hardware/software mapping (mapping of sub-systems to processes and computers, communication between computers)

This product does not need specific hardware. If the user has Google Chrome as a browser and a host to provide the database hosting, it can start using the extension. An Internet connection to visit BitBucket is also quite helpful, since the extension is used to look at pull request behaviour on BitBucket.

The computers only need a connection to the Internet to communicate with the database and BitBucket. If the database is hosted inside an organization, the computers do not need to connect to an external database. An Internet connection is not optional however, since a connection to BitBucket is still needed.

Persistent data management, database design



Concurrency (processes, shared resources, communication between processes, deadlocks prevention)

Not yet available

Glossary

Not yet available