1st Sprint – Web Application "Sami Rooms"

Benyamin Yakobi 323492835  
Sagi Vaknin 316605617  
Liron Vaknin 204447882   
Kesem Even-Hen 208055483  
Dan Marinescu 204264543

1. **Project Documentation**
   1. Functional Requirements
      1. All databases should be backup after certain amount of time in order to ensure a fail-safe for the system in case of a corruption in the database
      2. Relevant databases should be updated after adding / removing information by the user through the website, or in case of new successful reservation by customers.

|  |  |
| --- | --- |
| Epic | User Stories |
| Backend Management |  |

* 1. Non-Functional Requirements
     1. **Security**
        1. Database protection – As we are dealing with client's personal information such as names, addresses, credit cards etc. the database should be secured to prevent from these data to leak.
        2. Users must change the permanent login password they receive after password change process immediately after successful login.
        3. Password length shall be of 6-10 characters including letters & numbers.
     2. **Performance**
        1. Fast I/O – To further user personal experience the system should be able to provide service quickly for the users, Under one second per interaction.
     3. **Scalability**
        1. System capacity – The system should be able to work with 500 users simultaneously without crashing.
     4. **Efficiency**
        1. The system shall be able to handle the entry of orders by customers at a minimum rate of 10 per second.
     5. **Durability**
        1. All databases must be backed up locally to prevent loss of data in case of malicious attack or fatal system crash.
        2. The system shall be stored on a well-known & reliable cloud platform to ensure as much as possible its durability.
     6. **Availability**
        1. The Online Payment System shall achieve 100 hours MTBF (mean time between failure).
        2. The online payment system shall be available 24 hours a day, 7 days a week.
        3. Unless the website is non‐operational, the system shall present a user with notification informing them that the website is unavailable.
  2. Epics
     1. EXPORT FROM CLUBHOUSE
  3. User-Stories
     1. EXPORT FROM CLUBHOUSE
  4. Test Cases
* Note: Both Epics & User-Stories can be found in our [ClubHouse](https://app.clubhouse.io/scekdsl/epics)!

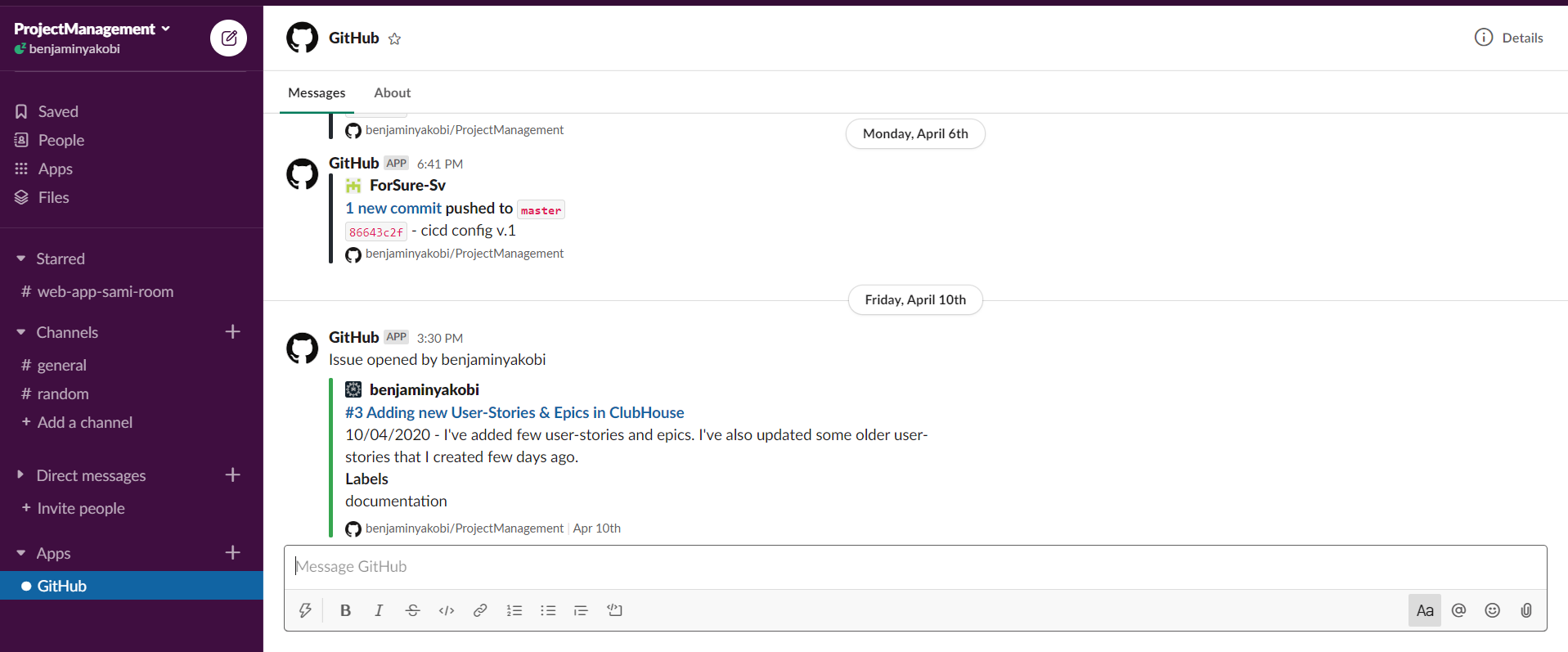
1. **Project Infrastructure**
   1. [GitHub Repository](https://github.com/benjaminyakobi/ProjectManagement) & Screenshot:

APPEND SCREENSHOT BEFORE ASSIGNMENT

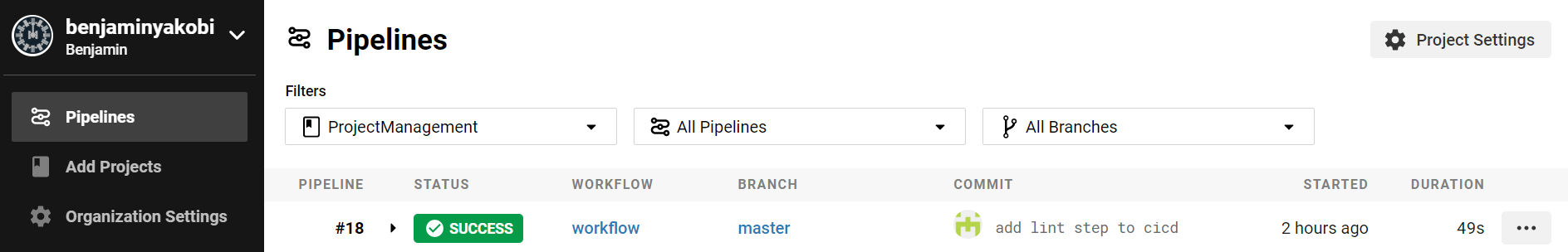
* 1. [Clubhouse Invitation](https://app.clubhouse.io/invite-link/5e7b729b-c807-4081-a089-08a3f44e208d) & Screenshot:

APPEND SCREENSHOT BEFORE ASSIGNMENT

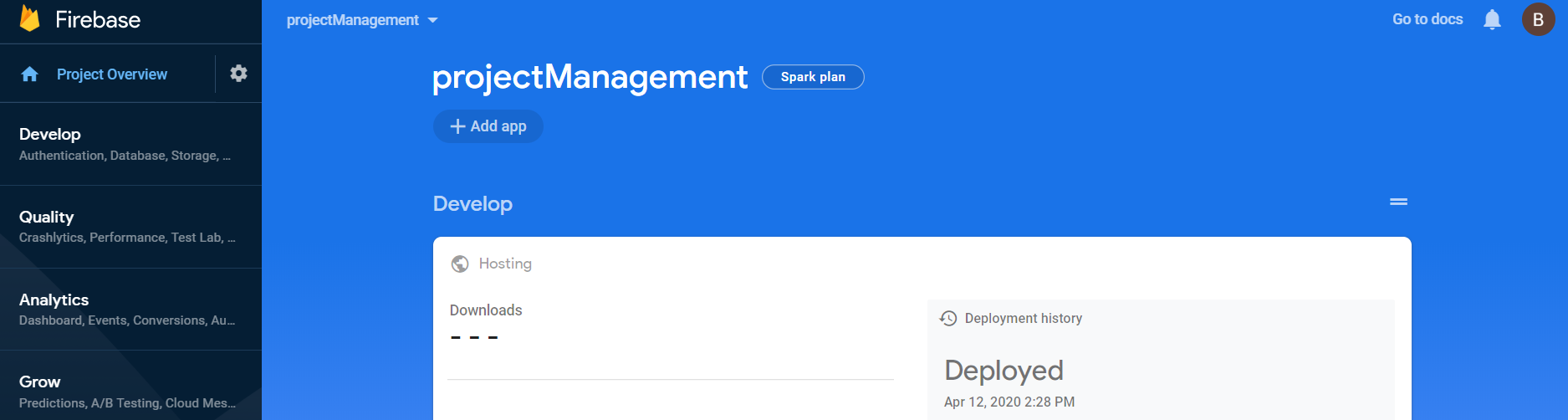
* 1. [Slack Invitation](https://join.slack.com/t/projectmanage-aem8080/shared_invite/zt-da9gfu1c-gmADLQBcDRu6s~aWyvI1aA) & Screenshot:

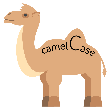


* 1. [CircleCi](https://app.circleci.com/pipelines/github/benjaminyakobi/ProjectManagement) & Screenshot:



* 1. [Firebase](https://console.firebase.google.com/u/0/project/projectmanagement-612b8/overview) & Screenshot:



* + In this project we will use Firebase as Cloud service to store & run the project.
  + In this project we will use Firebase as our DB to save our data for the next sprints.
  1. Convention – In this project we will use the 'camelCase' Convention.