

POLITECNICO DI MILANO



POLITECNICO
MILANO 1863

DESIGN AND IMPLEMENTATION OF MOBILE APPLICATIONS



Cracker

Design Document

by

Leonardo Febbo

Elisabetta Ferreri

February 17, 2020

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Definitions and acronyms	3
1.2.1	Definitions	3
1.2.2	Acronymns	3
1.3	Scope	3
1.4	Framework	4
1.5	Functional requirements	4
1.6	Non-functional requirements	4
1.7	Assumptions, dependencies, constraints	4
1.7.1	Assumptions and dependencies	4
1.7.2	Constraints	5
2	Architecture	6
2.1	Overview	6
2.2	High level components	6
2.3	Component view	6
2.4	Deploying view	6
2.5	Component interfaces	6
3	External Services and Libraries	7
4	Use Cases	8
5	Sequence Diagrams	9
6	User Interface Design	10
7	Testing	16
8	Used Tools	17

1 Introduction

1.1 Purpose

The purpose of this document is to describe the design phases of the realization of the "Cracker" mobile application, with particular attention to the architecture and the user experience.

The main aim of "Cracker" is to help users who read Marvel comics by letting them mark the issues they have read and keeping them up to date with the newest releases by Marvel. This project is the result of the implementation of the knowledge acquired during the course "Design and Implementation of Mobile Applications" provided by the Milan Polytechnic.

1.2 Definitions and acronyms

1.2.1 Definitions

- **Framework:** reusable set of libraries or classes for a software system
- **Issue:** a single, periodical, magazine-like publication
- **REST:** a way of providing interoperability between computer systems on the Internet
- **Series:** a title collecting a variable amount of issues
- **User:** a person who has performed the login operation successfully

1.2.2 Acronyms

- **API:** Application Programming Interface
- **IDE:** Integrated Development Environment
- **HTTPS:** HyperText Transfer Protocol Secure
- **JSON:** JavaScript Object Notation
- **MVC:** Model - View - Controller
- **URL:** Uniform Resource Locator

1.3 Scope

Cracker has been developed for users who regularly read Marvel comics and want to keep track of the series they are reading and the new issues that are released every week by the American publishing house. Similar apps already exist for other types of media, such as tv shows and movies, but none of them have comics as their focus. Keeping the possible needs of the users in mind, six main screens have been found of interest for our application:

- **Issue:** screen displaying information about a selected issue
- **Series:** screen displaying information about a selected series
- **Up Next:** list of issues to be published in the current week, in the next week and in the current month
- **To Read:** list of the next issues to read for each of the series the user follows
- **Search:** search for a series given its name
- **Profile:** personal information about the user

1.4 Framework

The development of Cracker was achieved through the use of native iOS SDKs, in particular by using the Swift programming language. This choice allowed greater control of system resources and access to system services, otherwise not possible if using cross-platform frameworks such as PhoneGap or React Native. The purpose is to implement different functionalities and integration with other sites.

1.5 Functional requirements

Cracker provides a simple, intuitive and user-friendly interface that allows the users to:

1. register with a personal e-mail or login with Facebook and Google
2. see information about an issue
3. see information about a series
4. mark series they are reading as *following*
5. select which issues of a series they have read
6. see the next issue not yet read for all the series they are following
7. see the series they are following
8. see statistics on the series they are following
9. see, on a weekly basis, the latest issues of all the series Marvel is publishing
10. search for a specific series, given its name

1.6 Non-functional requirements

The application must be able to:

- run on both iPhone and iPad
- adapt its written content to the language selected in the device settings (available languages: English, Italian)
- adapt to different screen sizes
- keep preferences and status at every start

1.7 Assumptions, dependencies, constraints

1.7.1 Assumptions and dependencies

- **Internet connection:** the device used by the user disposes of an Internet connection and a sufficient bandwidth to use the application
- **API availability:** the API provided by third part's services are always available
- **No privileged users:** there are no privileged users or administrators with particular functions
- **No user connections:** every user is independent from the others

1.7.2 Constraints

- **Hardware limitations:** our application runs on every mobile device like smartphones and tablets; therefore, as the app consumes a low amount of RAM, the only hardware constraint for the users is to have a mid-range device
- **Parallel operations:** the application must be able to handle multiple parallel requests with high reactivity

2 Architecture

2.1 Overview

2.2 High level components

2.3 Component view

2.4 Deploying view

2.5 Component interfaces

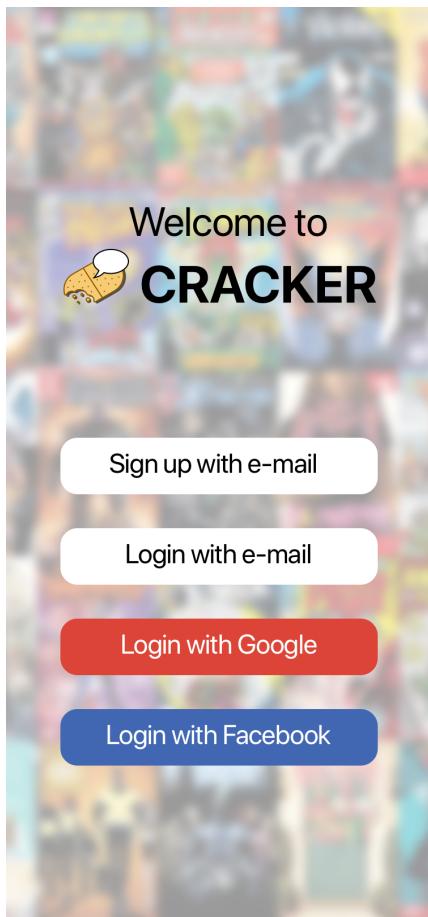
3 External Services and Libraries

4 Use Cases

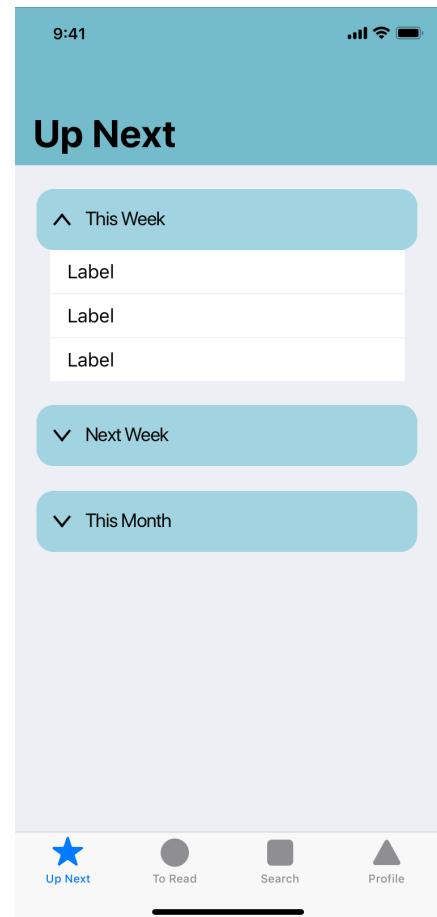
5 Sequence Diagrams

6 User Interface Design

In this section we show the user interfaces of all the screens of the application.

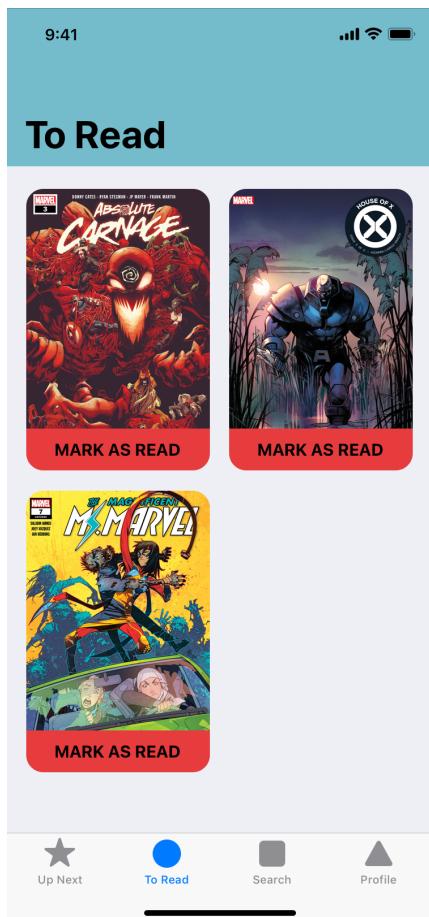


(a) Login

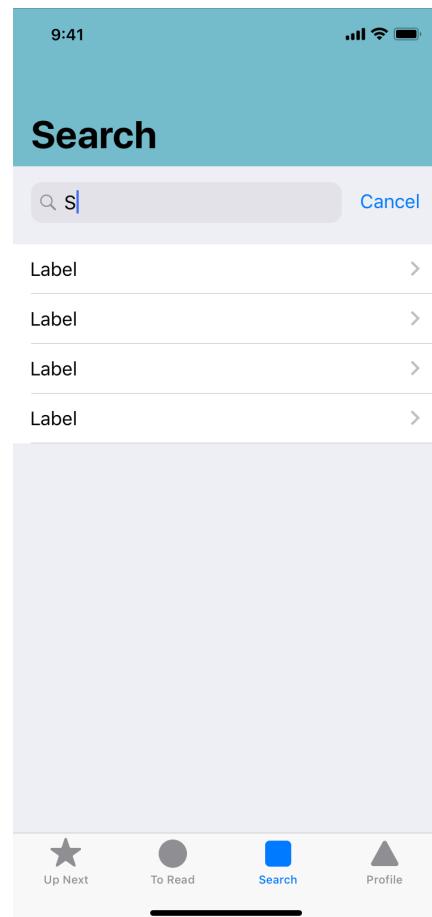


(b) Up Next

Cracker by Leonardo Febbo, Elisabetta Ferreri

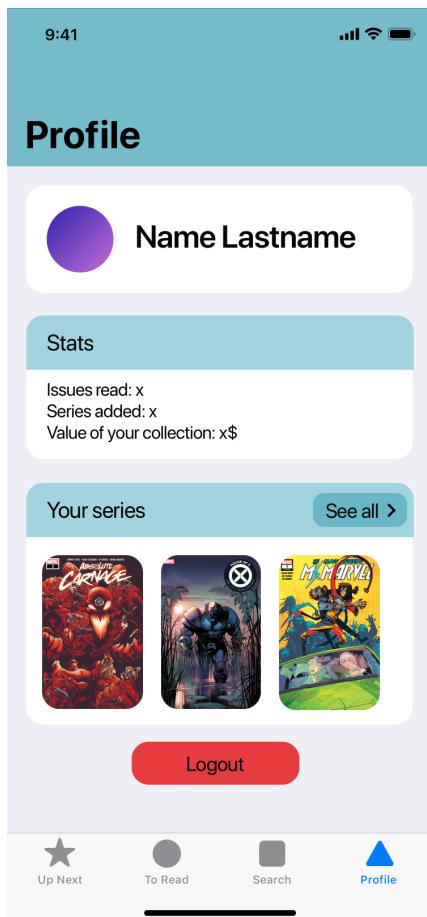


(a) To Read

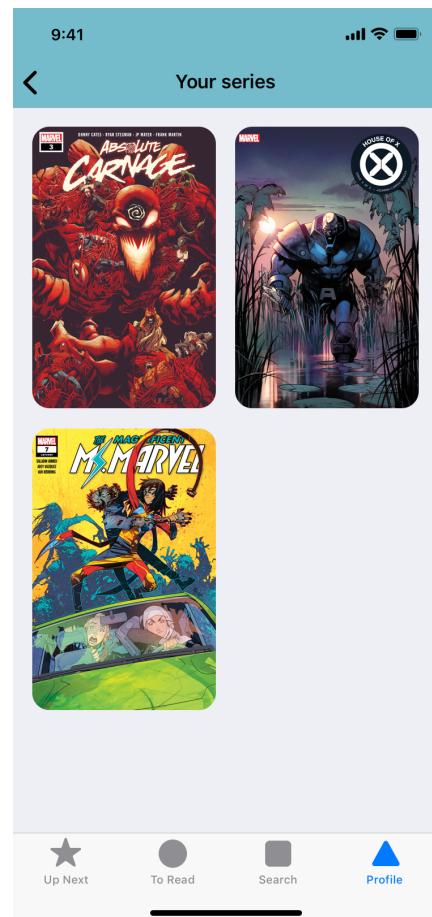


(b) search

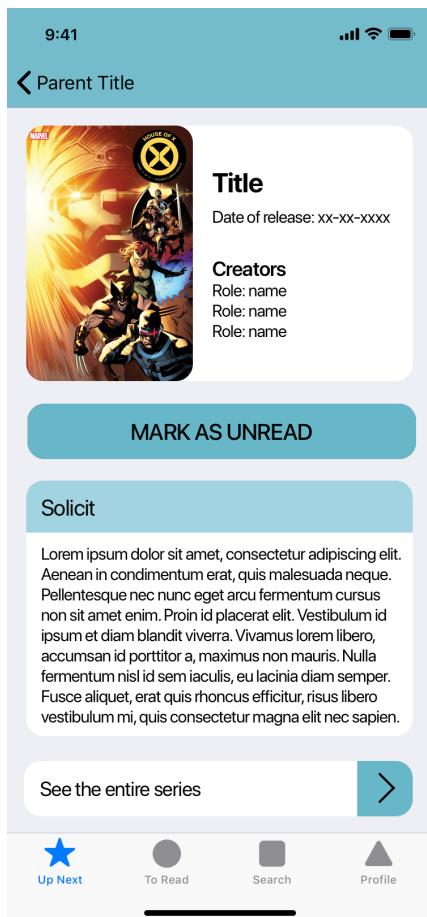
Cracker by Leonardo Febbo, Elisabetta Ferreri



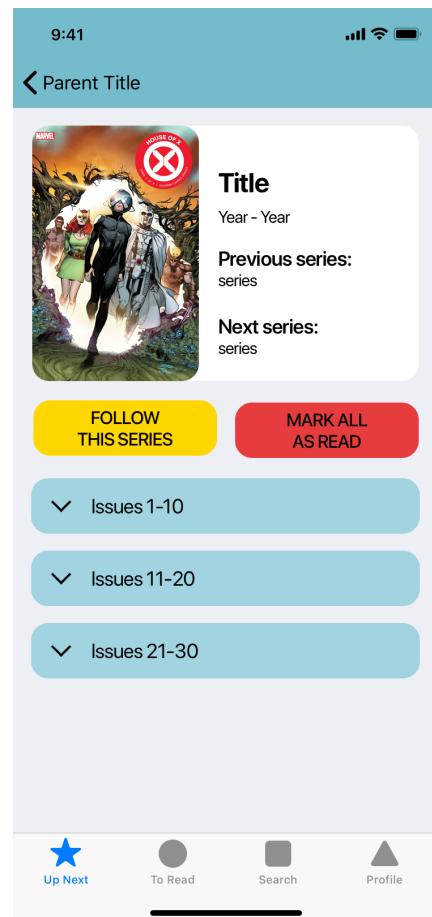
(a) Profile



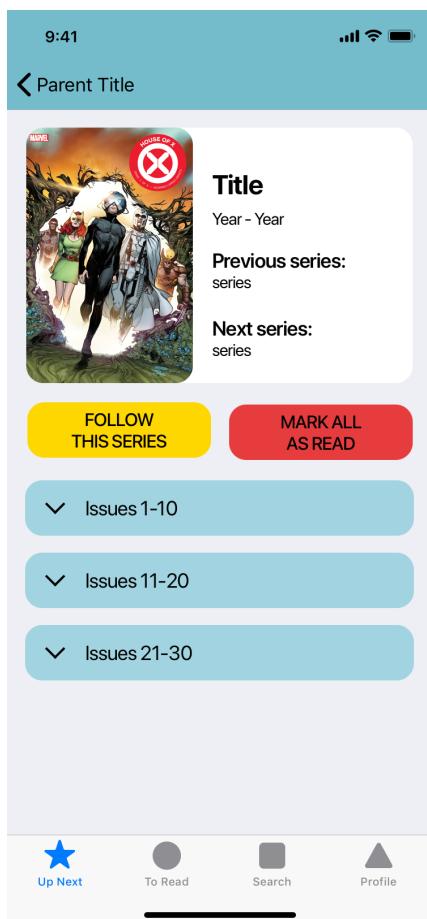
(b) Your series



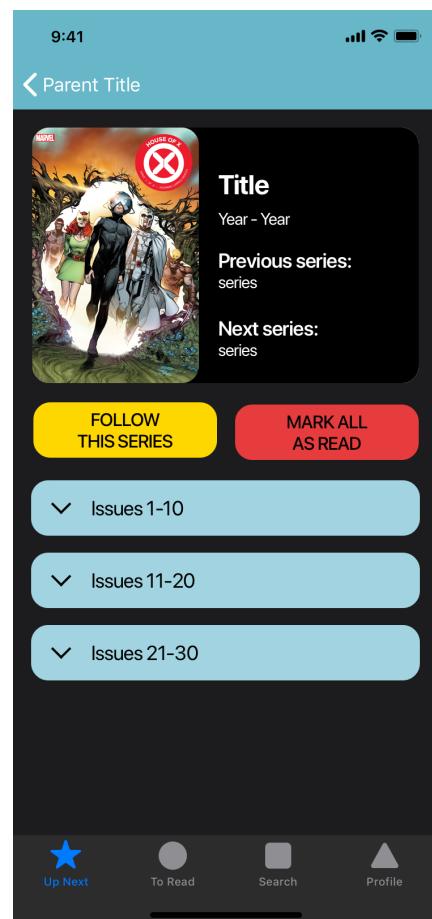
(a) Issue



(b) Series

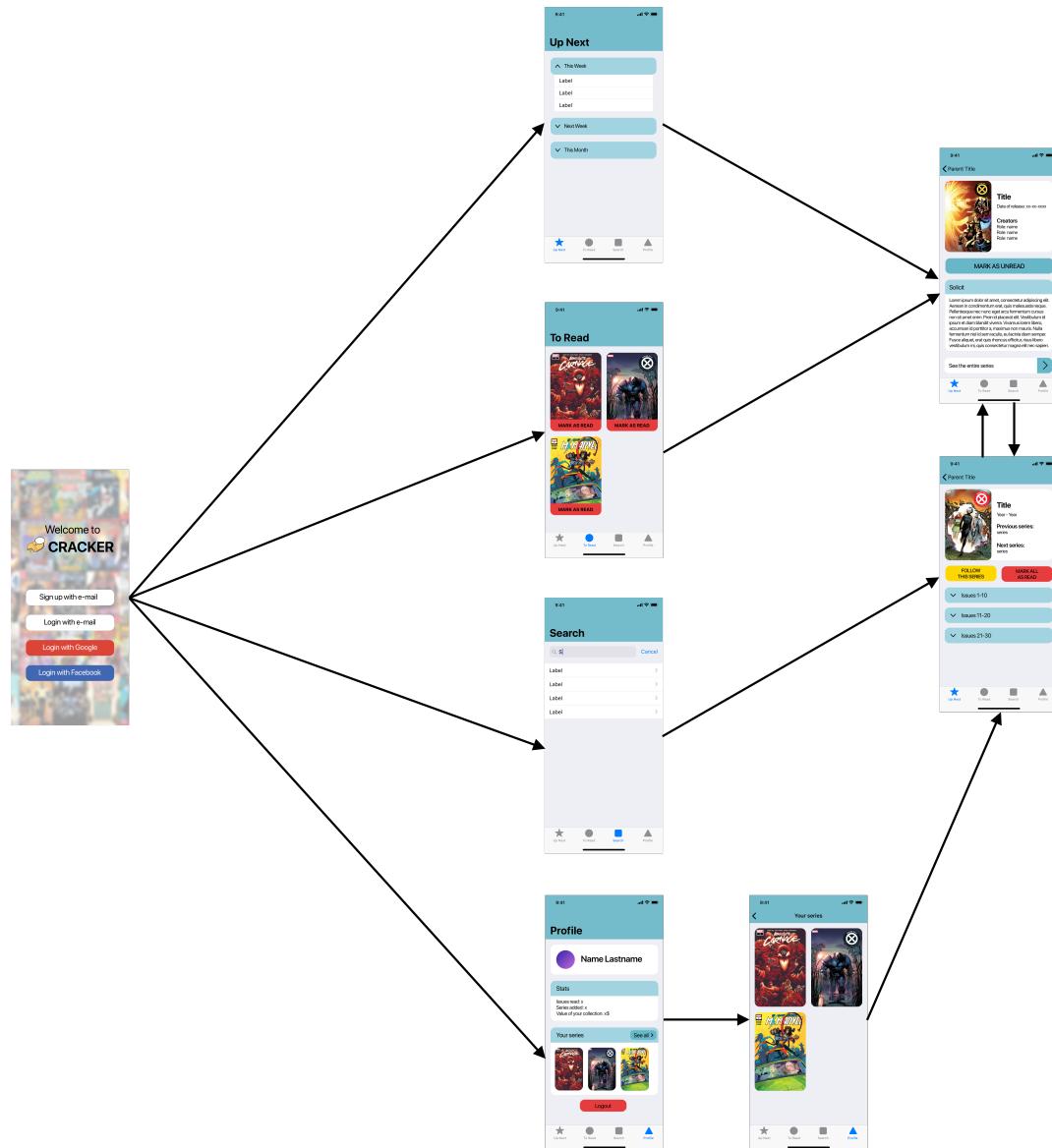


(a) Series: light mode



(b) Series: dark mode

We close this section by showing the interaction between the different screens.



Cracker by Leonardo Febbo, Elisabetta Ferreri

7 Testing

8 Used Tools

The following softwares, tools and services have been used for the development of the application:

- XCode 11.3
- Sketch
- Github
- Firebase