Title: Some kind of mobile application for Spain.

Additional points :

- List of User Stories should be a part of strategy.
- Each User Story should contain a device or several devices where it may be tested.
- We do regression testing each 3-4 sprints. In other sprints - detailed Smoke test.

Backend part:

Test related API methods in advance of the current sprint (or at the very beginning of the sprint)

Tools: ***Postman, Charles***

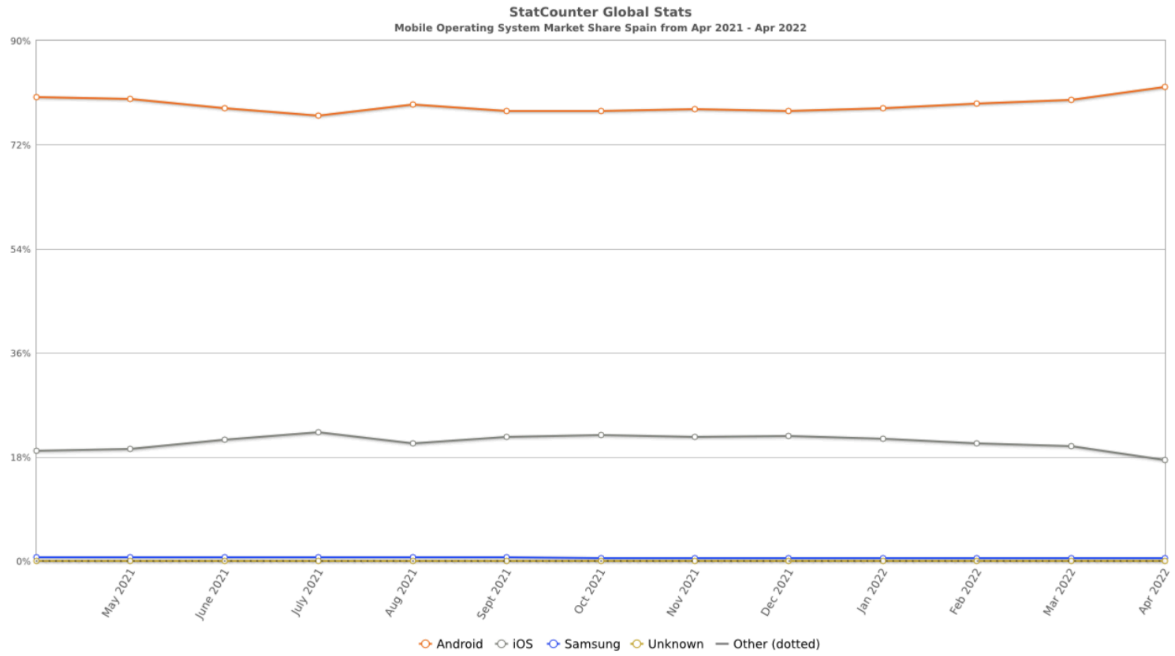
Frontend part (mobile application):

List of supported devices

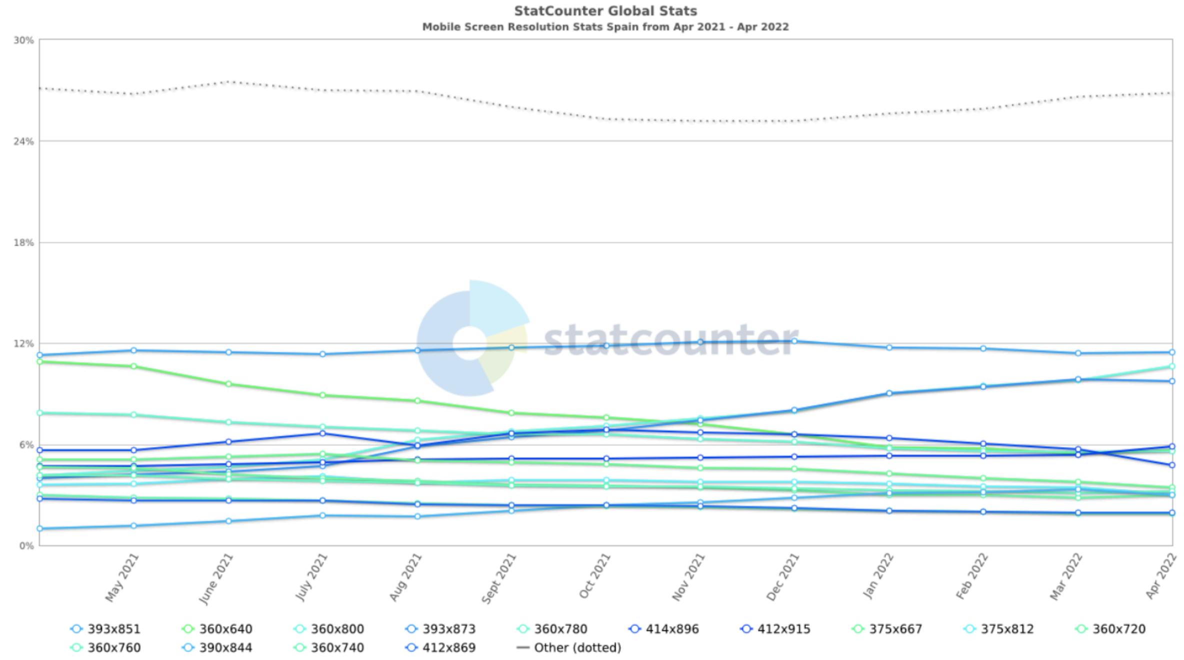
Device	Type of testing	Version	Display	Screen resolution
iPhone XR	regression	14.4	6.1 inch	828 x 1792 (viewport 414 x 896)
iPhone 11	smoke	14.5	6.06 inch	1792 x 828 (viewport 390 x 844)
Samsung Galaxy 10	smoke	10	6.1 inch	3040×1440 (viewport 360 x 760)
Samsung Galaxy S9	regression	10	5.8 inch	1440x2960 (viewport 360x740)
OnePlus 7 pro	regression	11	6.67 inch	3120x1440 (viewport 412 x 892)
Huawei P20 Lite	smoke	10	5.84 inch	1080x2280 (viewport 360x760)

For IOS support two latest versions.
Android supports four latest versions.

Mobile Operating System Market Share Spain (Apr 2021 - Apr 2022)



Mobile Screen Resolution Stats Spain (Apr 2021 - Apr 2022)



Test preparation

Write **Test cases** for **User stories** from requirement doc.

Update existing test cases during the development process in case things are changing.

*Tools: **TestRail***

Prepare required test environment

Browsers, environments, PC, devices, utilities, test-data

*Tools: **Browserstack, Android Studio, XCode***

Perform testing

Record results to **TestRails** and **bug tracking system**. Verify Bugfix.

*Tools: **TestRail, JIRA***

Product delivery

Delivery the product with no critical issues at the end of each sprint, release notes to be attached. The delivered build may contain hard-to-reproduce, device-specific or trivial issues.

*Tools: **FireBase App Distribution***

Testing starting Criteria

- test documentation is ready;
- related functionality is implemented and merger reviewed by developers;
- a build is provided;
- a test environment is ready.

Types of test activities

- **Test Strategy preparation** - to know what, when, and how to test;
- **Test Designing** - to determinate general approaches for test cases creation;
- **Test Cases creation** - to follow them during testing the application;
- **Exploratory testing** - to form a general impression of the build and provide quick feedback;
- **Functional testing** - to make sure the features work as described in requirements all over the supported devices and "operational systems";
- **Acceptance testing** - to make sure the system corresponds to acceptance criteria;
- **Validation testing** - to make sure the feature really meets business and user needs;

- **Compatibility testing** - to make sure the app works correctly on all browsers/devices claimed;
- **Usability testing** - to make sure the application meets a user-friendly UX;
- **Smoke testing** - to make sure there are no critical issues before sprint delivery. Activity is permissible in case some considerable changes are being made(e.g. refactoring, adding new features which may impact delivered functionality);
- **Regression testing** - to make sure there are no critical issues before final product delivery. Activity is permissible in case of some significant, vital changes that affect the functionality of the whole app(performance improvements, changes in a whole microservice(like creating new or deleting/replacing existed), changes in the response body/status codes). Regression will be executed once in 3 Sprints.

Additional types of testing may take place in case of need.

Definition of Done and Exit Criteria

- All users stories are tested;
- All critical test cases are passed;
- All founded defects are reported;
- Regression/smoke testing before delivery is held;
- No Blockers / Critical-severity defects in **Open** Status;
- 2 Normal-severity defects are in **Open** status;
- 5 low-severity defects are in **Open** status;
- Client is informed about all defects which will go with release;
- Testing was done in a production-like (Stage) environment.

Production level ready solution must meet all definition of done and exit criteria

Defects reporting and management

All test cases that were marked as ‘failed’ should be reported to the project JIRA board with proper severity and priority.

4 types of issue severity will be used:

Severity	**Description**

Blocker	Are issues that can prevent a user from using the application or its core features and it does not have a workaround. Examples of the issues that can be reported as 'Blockers': User is blocked to solve business needs the app should help to solve, e.g.: ● user can not login into the application; ● application crashes after login.
Critical	Are issues that affect critical functionality or critical data that users use most of the time. Almost all crashes that can be reported are created with this priority. Examples of the issues that can be reported as 'Critical': Core functionality extension is not available, e.g.: ● user can not pair device ● user can not choose settings
Normal	Are issues that affect minor functionality or non-critical data. Such issues usually have an easy workaround. Examples of the issues that can be reported as 'Normal': There are some specific conditions when additional features or Core functionality extensions are not available, e.g.: ● Medium UI issues (different size/colors of fonts or icons, some elements a little bit shifted); ● validation rules issues.
Low	Are issues that do not affect functionality or data. Also, we reported rarely reproduced specific crashes with this severity: ● there is some sequence of user actions that may lead to UI or logical issues what do not affect core features or business logic of the application; ● hardly notable UI artifacts.

Risks

- Impossible to start testing of the implemented functionality due to blockers on the part of developers from the client side;
- A significant number of "change requests" during development;
- Do not start testing the functionality that we know will most likely be changed;
- Do not start regression testing until client approves it was the last change request;
- Accuracy and completeness of documentation for testing: requirements / api methods / scenarios;
- Completeness of testing time;
- Internal team stability;
- Planning errors;
- Complicated animations, screen transitions, or complex gestures in mobile app.