|  |
| --- |
| A30 – Concevoir et développer des applications mobiles |

Documentation du projet « SpaceX Launches »



Alexis Stella

Version 1 du 15 juin 2021

Module du 17 mai 2021 au 15.06.2021

Table des matières

[1 Introduction du module 3](#_Toc74642134)

[2 Introduction « SpaceX Launches » 4](#_Toc74642135)

[2.1 Plateforme de développement 4](#_Toc74642136)

[2.2 Synopsis 4](#_Toc74642137)

[3 Analyse et conception 4](#_Toc74642138)

[3.1 Description des fonctionnalités souhaitées 4](#_Toc74642139)

[3.2 Maquettes 5](#_Toc74642140)

[3.2.1 Accueil 5](#_Toc74642141)

[3.2.2 Inscription 5](#_Toc74642142)

[3.2.3 Connexion 6](#_Toc74642143)

[3.2.4 Liste des lancements 7](#_Toc74642144)

[3.2.5 Informations du lancement 7](#_Toc74642145)

[3.3 Tests technologiques 9](#_Toc74642146)

[4 Implémentation 10](#_Toc74642147)

[4.1 Problèmes rencontrés 10](#_Toc74642148)

[4.1.1 Problème d’affichage de components 10](#_Toc74642149)

[4.1.1.1 Définition 10](#_Toc74642150)

[4.1.1.2 Solution 10](#_Toc74642151)

[4.2 Résultat obtenu 11](#_Toc74642152)

[4.2.1 Accueil 11](#_Toc74642153)

[4.2.2 Log In 11](#_Toc74642154)

[4.2.3 Register 12](#_Toc74642155)

[4.2.4 Liste lancements 13](#_Toc74642156)

[4.2.5 Information lancement 15](#_Toc74642157)

[5 Tests 16](#_Toc74642158)

[5.1 Tests fonctionnels 16](#_Toc74642159)

[6 Conclusion 18](#_Toc74642160)

[6.1 Ce que vous avez aimé 18](#_Toc74642161)

[6.2 Ce que vous n’avez pas aimé 18](#_Toc74642162)

[6.3 Auto-évaluation 18](#_Toc74642163)

[7 Sources 19](#_Toc74642164)

# Introduction du module

Dans le module A30, nous allons apprendre à modéliser l’expression des besoins sous forme d’une analyse des besoins. Nous allons aussi apprendre à appliquer le modèle de conception et réaliser l’implémentation de l’application en respectant les prescriptions, les conventions et en utilisant les spécificités des applications mobiles.

Pour terminer, nous allons apprendre à tester l’application de manière fonctionnelle en se basant sur l’analyse des besoins et en utilisant le « débuggeur ». Nous allons ensuite apprendre à établir la documentation exhaustive.

# Introduction « SpaceX Launches »

## Plateforme de développement

L’application de ce projet utilisera l’outil « CodeName One » qui est un plugin disponible pour IntelliJ, Eclipse et NetBeans. J’ai voulu choisir « CodeName One » car ce plugin permet de faire du développement d’application multiplateformes véritablement natives avec Java pour des appareil iOS, Android ou du Web.

## Synopsis

L’application « SpaceX Launches » sera une application sur mobile qui permettra de recevoir les dernières nouvelles sur le prochain lancement ainsi que les anciens lancements de l’entreprise SpaceX. Le visiteur de cette application devra se connecter pour pouvoir afficher les différents lancements ainsi que leurs informations. Il pourra, s’il ne dispose d’aucun compte utilisateur, s’inscrire et ainsi se connecter.

Dès que l’utilisateur est connecté, il pourra voir le prochain lancement ainsi que les anciens lancements en fonction de leur date. Il pourra, si cela se fait, demander de mettre un rappel pour le prochain lancement. En appuyant sur un lancement, il sera possible de voir les informations de celui-ci : numéro de vol, nom de la mission, date du lancement, nom de la fusée, … L’utilisateur pourra aussi trier les lancements qu’il veut voir. Par exemple si un utilisateur veut voir les vols par date croissante ou décroissance, en fonction du nom (aussi par ordre croissant décroissant). Ce paramètre de trie sera automatiquement enregistré dans le cache de l’utilisateur.

# Analyse et conception

## Description des fonctionnalités souhaitées

Cette application aura beaucoup de fonctionnalités pour pouvoir subvenir au synopsis. Nous avons mis les fonctionnalités du synopsis dans la version 1 de notre application. Nous avons aussi imaginé d’autres fonctionnalités nous pouvions ajouter dans de future mise à jour/implémentation.

Voici un tableau regroupant ces fonctionnalités ainsi que leur version :

|  |  |
| --- | --- |
| Fonctionnalité | Version |
| Connexion via un email et un mot de passe. | Version 1 |
| Inscription via un nom, prénom, email et mot de passe. Vérification de l’email pour voir s’il existe déjà dans la DB. |
| Appel d’une api pour les lancements de SpaceX. Voici l’url de l’api que nous allons utiliser 🡺 https://github.com/r-spacex/SpaceX-API |
| Affichage des lancements à la connexion de l’utilisateur ou au moyen du bouton refresh. |
| Tri des lancements en fonction de leur date/nom/numéro de vol |
| Enregistrement du paramètre de tri dans le cache du mobile de l’utilisateur. |
| Barre de recherche pour rechercher un lancement. Ce sera une recherche fulltext sur le contenu. | Version 2 |
| Enregistrement du filtre dans une base de données |
| Amélioration de l’interface utilisateur. | Version 3 |
| Système de rappelle du prochain lancement (avec une notification) |

## Maquettes

Voici les maquettes de l’application :

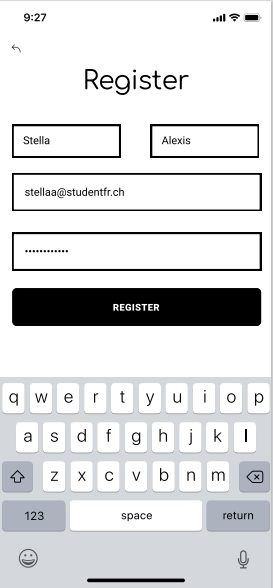
### Accueil

La vue accueil contiendra une image de fond avec le logo de SpaceX. En bas de cette vue, nous aurons deux boutons ; un pour se connecter et un autre pour s’inscrire.



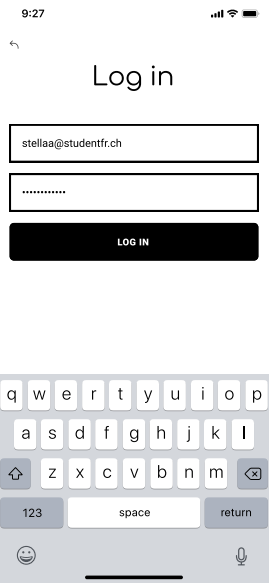
### Inscription

Dans la vue inscription, nous aurons tout d’abord en haut, un bouton pour revenir sur la page d’accueil. Ensuite nous aurons un formulaire qui demandera le nom, prénom, email et le mot de passe. Pour terminer, un bouton « Register » sera là pour pouvoir s’inscrire avec les informations que nous avons précédemment rentrées.



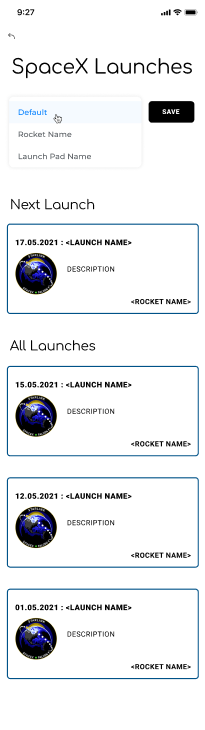
### Connexion

Dans la vue connexion, nous aurons tout d’abord en haut, un bouton pour revenir sur la page d’accueil. Ensuite nous aurons un formulaire qui demandera l’email et le mot de passe. Pour terminer, un bouton « Log In » sera là pour pouvoir se connecter avec les informations que nous avons précédemment rentrées.



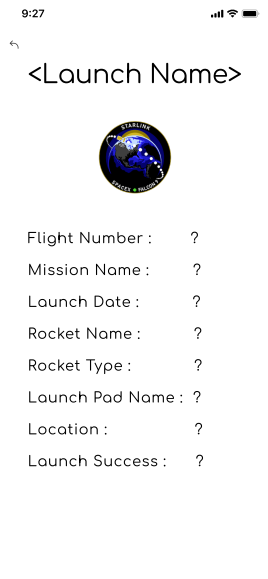
### Liste des lancements

Cette vue contiendra tous les lancements passés ainsi que le prochain lancement. En haut de cette vue, un bouton pour revenir sur la page d’accueil sera là. Il servira aussi comme bouton de déconnexion. Pour terminer, une ComboBox sera là pour pouvoir trier les lancements



### Informations du lancement

En cliquant sur un lancement, une nouvelle vue sera affichée. Cette vue contiendra des informations supplémentaires sur le lancement. Un bouton de retour vers la page des lancements sera en haut de cette vue.



## Tests technologiques

Pour réaliser notre application mobile, nous devons utiliser des fonctionnalités que nous devons d’abord tester avant de commencer.

Voici une liste de ces tests technologiques ainsi que leur but :

|  |  |
| --- | --- |
| Tests utiles | Buts |
| Structure de codename one | Le but de ce test est de voir/mettre en place l’infrastructure fournie par codename one via le plugin pour IntellijJ. Ce test est très important car il nous permet de tout de suite détecter s’il y a des problèmes avec cette infrastructure et ainsi ne pas les avoir durant l’implémentation. |
| Hébergement sur Tizoo | Ce test a pour but de tester si l’hébergement de services REST fonctionne sur Tizoo. Cela nous permettra de tout de suite se rendre compte si cette technologie fonctionne car nous en aurons besoin lors de l’implémentation. |
| Envoyer une version Build pour Android | Ce test a pour but de tester si nous arrivons a correctement envoyé un build du projet de base de codename one pour vérifier si tout fonctionne correctement. Cela nous permettra de savoir si nous avons aucun problème pour build un projet basique et ainsi réduire le champ de recherche en cas d’erreur. |
| Requête REST (Méthode) | Ce test a pour but de tester si sur codename one il est possible de facilement faire des requête REST vers une api par exemple. L’idée est que nous allons faire un server REST donc ce test technologique est très important pour le bon déroulement de la suite. |
| Requête vers notre serveur REST | Le but dans ce test est de voir si nous arrivons avec la méthode de codename one, de faire des requêtes vers un serveur REST que nous avons créé pour le test. L’idée est de tester si nous arrivons à faire des requêtes vers un serveur hébergé sur Tizoo et ainsi voir si le serveur renvoie bien les informations. |
| Lecture de JSON | Dans ce test, nous allons tester la lecture d’un JSON reçu lors d’une requête. L’idée est de voir comment il est possible de lire ce JSON car c’est ce format que nous allons utiliser pour notre serveur REST. |
| Base de données | Le but de ce test est de tester l’hébergement d’une base de données sur notre compte Tizoo. L’idée est que nous allons en utiliser une pour implémenter notre application donc le but est de trouver des problèmes/régler des problèmes pour ne pas les avoir durant notre implémentation. |

# Implémentation

## Problèmes rencontrés

### Problème d’affichage de components

#### Définition

Lors de l’implémentation de mon application mobile, j’ai rencontré un problème lors de l’ajout de nouveaux components. Le problème était que ces components ne s’affichaient pas lorsque je les ajoutais au Form principal. Le souci était donc que je ne voyais qu’un écran blanc au lieu de voir tous les components, la liste des lancements par exemple.

#### Solution

Pour résoudre ce problème, j’ai tout simplement fait l’appel à la méthode « revalidate » comme en Swing. Cela permet de réactualiser le container et ainsi mettre à jour avec les nouveaux components.

Voici un exemple qui provient de mon code.

add(list);

revalidate();

## Résultat obtenu

### Accueil

Voici une image du résultat de la vue de l’accueil de notre application :

A picture containing text, monitor, mirror, car mirror

Description automatically generated

Comme nous pouvons le voir, nous avons respecté la maquette que nous avions réalisé. Nous pouvons alors cliquer sur « Log In » pour se rendre sur la page de connexion et « Register » pour s’inscrire.

### Log In

Voici maintenant une image du résultat de la vue de connexion. Sur cette vue, nous n’avons rien changé par rapport à la maquette.

Graphical user interface, application

Description automatically generated

Voici un exemple d’erreur sur cette vue :

Graphical user interface, application

Description automatically generated

### Register

Voici une image du résultat de la vue d’inscription. Sur cette vue, nous n’avons rien changé par rapport à la maquette comme pour la connexion.

Graphical user interface, application

Description automatically generated

Voici un exemple d’erreur sur cette vue :

Graphical user interface, application

Description automatically generated

### Liste lancements

Voici une image du résultat de la vue qui contient la liste des lancements. Sur cette vue, les seules choses qui ont changé est que nous n’utilisons pas de combobox comme nous l’avions prévu sur notre maquette. Le fait que nous n’utilisons pas de combobox est que nous avons utilisé un InfiniteContainer et que c’est très compliqué de mettre une combobox. Nous avons décidé de partir sur un bouton qui affichait un modal lorsque nous cliquons dessus. La deuxième chose est quand nous n’utilisons d’image sur chaque lancement de la liste contrairement à la maquette. La raison est que cela demandait trop de requête depuis le client et que ce n’était pas viable pour une situation réelle.

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

### Information lancement

Voici une image du résultat de la vue qui contient les informations sur un lancement. Sur cette vue, nous n’avons rien changé par rapport à la maquette.

Table

Description automatically generated

# Tests

## Tests fonctionnels

Voici la liste des résultats de tests fonctionnels effectués le 15.06.2021 :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N° | Description | Résultat attendu | Résultat obtenu | OK ? |
| 1 | Un utilisateur connecté peut visualiser tous les lancements. | L’utilisateur a bien accès aux lancements lorsqu’il est sur la vue « Logged » | L’utilisateur a bien accès aux lancements lorsqu’il est sur la vue « Logged » | OK |
| 2 | Un visiteur peut s’inscrire via son nom, prénom, email et son mot de passe. | Le visiteur peut s’inscrire ce qui impacte la base de données. Il peut alors, se connecter. | Le visiteur peut s’inscrire ce qui impacte la base de données. Il peut alors, se connecter. | OK |
| 3 | Un visiteur disposant d’un compte peut se connecter si ces identifiants sont corrects. | Le visiteur peut se connecter via son email et son mot de passe. Il a alors accès à la liste des lancements. | Le visiteur peut se connecter via son email et son mot de passe. Il a alors accès à la liste des lancements. | OK |
| 4 | Un utilisateur peut se déconnecter ce qui termine sa session. | L’utilisateur peut bien se déconnecter, sa session est correctement terminée. Il est ensuite redirigé vers la vue accueil | L’utilisateur peut bien se déconnecter, sa session est correctement terminée. Il est ensuite redirigé vers la vue accueil  (il est aussi déconnecté de la base de données) | OK |
| 5 | Un visiteur essaie de se connecter avec un mauvais email et mot de passe. | Un message d’erreur est bien affiché et le visiteur reste non-connecté. | Un message d’erreur est bien affiché et le visiteur reste non-connecté. | OK |
| 6 | Un visiteur essaie de s’inscrire mais n’a pas rempli tous les champs. | Un message d’erreur est affiché et le visiteur n’est pas inscrit. | Un message d’erreur est affiché et le visiteur n’est pas inscrit. | OK |
| 7 | Un utilisateur connecté peut trier les lancements en cliquant sur le bouton de tri. | Les lancements sont correctement triés en fonction du mode de tri sélectionné. | Les lancements sont correctement triés en fonction du mode de tri sélectionné. | OK |
| 8 | Enregistrement du paramètre de tri dans le cache du mobile. | Lorsque l’utilisateur change le paramètre de tri et relance l’application, le paramètre de tri est alors pris du cache. | Lorsque l’utilisateur change le paramètre de tri et relance l’application, le paramètre de tri est alors pris du cache. | OK |

# Conclusion

En conclusion, dans le module A30, nous avons appris à modéliser l’expression des besoins sous forme d’une analyse des besoins. Nous avons aussi appris à appliquer le modèle de conception et réaliser l’implémentation de l’application en respectant les prescriptions, les conventions et en utilisant les spécificités des applications mobiles.

Pour terminer, nous avons appris à tester l’application de manière fonctionnelle en se basant sur l’analyse des besoins et en utilisant le « débuggeur ». Nous avons ensuite appris à établir la documentation exhaustive.

## Ce que vous avez aimé

Dans ce module, j’ai beaucoup aimé le fait de faire une application mobile. L’idée d’avoir changé des applications basiques que nous réalisions m’a beaucoup plus. J’ai aussi beaucoup aimé le fait que c’était sous forme de projet personnel. En effet, j’ai pu réaliser une application dont j’aurai l’utilité dans le futur tout en ayant appris beaucoup de choses. J’ai aussi beaucoup aimé la liberté que nous avions durant ce module, c’était très motivant et me boostait énormément.

## Ce que vous n’avez pas aimé

Dans ce module, j’ai moins aimé le fait que nous n’avions pas les 5 blocs comme d’habitude dû aux évaluations de maturité. Sinon je n’ai rien à déclarer et je recommande fortement ce module !

## Auto-évaluation

Durant ce module, j’ai choisi de partir sur CodeName One pour réaliser mon application. Étant du JAVA, je n’ai pas appris énormément de choses, par conséquent, la façon dont fonctionne CodeName One m’a permis de me creuser la tête à certains moments. En effet, pour pouvoir build pour mobile, nous devons souvent modifier du code JAVA car le compilateur ne comprend pas certaines choses des nouvelles versions de JAVA. J’ai aussi beaucoup appris sur les certificats (IOS) et le casse-tête que c’est, dans le futur je partirai probablement sur une PWA car ça reste quand même l’avenir. Toutefois cela m’embête un peu car j’aime beaucoup JAVA.

# Sources

<https://www.codenameone.com/>

https://www.codenameone.com/developing-in-codename-one.html