**Schmitz-Dynamics**

**Schmitz-Dynamics**

Bernstrasse 109a

3052 Zollikofen

Telefon: +41 78 927 72 56

Modern informationstechnology

Zollikofen, 03. November 2016

**Cost-Estimate Breguet**

Foreword

This cost estimate is based on the document “Requirements Breguet Online Marketing Platform v1.2.pdf” and questions asked during the meeting of the 01 November 2016.

Other content related to the project which was not described in the RE Document was not considered in this estimate.

Content

The scope of work was cut down in tasks to achieve all defined requirements and to satisfy the customer needs.

Corresponding tasks were consolidated into task types. Following 10 types were defined:

* Administrative
* Preparation
* Phase 1
* Phase 2
* Phase 3
* Phase 4
* Phase 5
* Workflow Beta and Alpha
* Release & Go Live
* Fixing DB Schema

Following pages will describe which features / pages and tasks are implemented in the task type to achieve the global goal of the Media Application for Breguet.

# Administrative

The main part of the administrative task is to communicate with the end customer and to keep up the documentation of the project and plan further steps. The aim is to minimise the programming effort to the minimum with thought-through preparation.

Following administrative tasks are defined:

|  |  |
| --- | --- |
| Task | Est. duration in hrs |
| Creating Scribbles | 6 |
| Creating Design Moqups | 12 |
| Writing Test-Concepts | 10 |
| Requirements Engineering (through the project) | 40 |
| Writing security access documentation | 6 |
| Creating a database schema | 12 |
| Creating administrative documents | 8 |

# Preparation

The preparation considers to set up a development and designing environment to grant fast and reliable programming. The preparation also includes a basic skeleton app with specific required components for the project. After the preparation process every custom component for the application should be created and reliable. This accelerates the further development tremendously.

Following preparation tasks are defined:

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Setup | Setting up design environment | 2 |
| Design | Navigation | 4 |
| Design | Search | 2 |
| Design | Modal | 2 |
| Design | E-Mail-Layout | 4 |
| Design | Basic CSS | 6 |
| Setup | Setting up programming environment | 2.5 |
| Setup | Setting up skeleton app | 4 |
| Setup | Create global component | 8 |
| Database | Setting up the database | 1.5 |

# Phase 1 (Users / Groups / Permissions)

The first phase of the project contains following features:

* Enables the user to login
* Users can be created by the media admins
* CRUD[[1]](#footnote-1) functionality for Users / Permissions / Groups
* A media admin account is created for testing purpose.
* A user can be added to a group
* A user can get the according permissions
* A user can reset his password

Following pages are created:

* Permission-Page
* Users-Page
* Login-Page
* User-Profile Page
* Reset-Password Page

Following tasks are required to get those features:

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Design | Permission - Page | 2 |
| Design | Users - Page | 2 |
| Design | Login - Page | 2 |
| Design | Reset - Password Page | 2 |
| Database | Create user schema | 0.75 |
| Database | Create permission Schema | 0.5 |
| Database | Create group Schema | 0.25 |
| Type-Definition | Create type definitions | 1.5 |
| CRUD | CRUD functionality for users | 7 |
| CRUD | CRUD functionality for permissions | 8 |
| Backend | Create API - users / permissions / groups | 3 |
| CRUD | CRUD functionality view groups | 2 |
| Testing | Writing Unittests Phase 1 | 4 |

# Phase 2 (Base-Data)

The phase two contains the CRUD functionality for following-database entities:

* Publication
* Layouts
* Producers
* Types
* Currencies
* Markets
* Regions

In addition, the corresponding and already existing data will be imported for the further use in the application.

Following pages will be created:

* Publication (new, edit, view, delete)
* Layouts (new, edit, view, delete)
* Producers (new, edit, view, delete)
* Types (new, edit, view, delete)
* Currencies (new, edit, view, delete)
* Markets (new, edit, view, delete)
* Regions (new, edit, view, delete)

Following tasks are required to get those features:

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Design | Design Base Data Pages (7 Pages) | 4 |
| CRUD | CRUD functionality Base - Data | 8 |
| Type-Definition | Create type definitions | 2 |
| Testing | Writing Tests Phase 2 | 2 |
| Backend | Create API - Base Data | 3 |
| Import | Import Base Data | 6 |

# Phase 3 (Excel - Import)

The third phase focuses on the excel import. After this phase Breguet should be able to:

* Upload their excels and generate a corresponding report
* Get visible validation output for their imports

Following tasks are required to get those features:

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Design | Excel-Import | 3 |
| Programming | Excel-Import functionality | 6 |
| Programming | Basic Job - Page Structure | 2 |
| Programming | Basic Service / Basic Controller | 2 |
| Testing | Unit-Tests | 4 |
| Programming | Excel-Calendar Import | 8 |

# Phase 4 (Job – Process)

The forth phase is the biggest phase and has the aim to allow the job creation from A-Z.

* All required Job-pages are created and a job can be generated manually or automatic via import.
* According to the current status of the job e-mails will be sent to the responsible person
* Permission management will be implemented.
* A task gets created respecting the job status

Following pages are created:

* Job list page
* Job approval page
* Create a job manually
* Different E-Mail layouts for job notifications
* Editing a job
* Job question page
* Job comment page
* Task-List page
* Task status

Following tasks are required to get those features:

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Design | Extend / adjust Job - Page Design | 1 |
| Design | Design all Job - Pages | 6 |
| Design | Create the E-Mail Design for Job's | 2 |
| Type-Definition | Writing schema and types | 5 |
| Services | Writing Services | 3 |
| Programming | Create Job Manualy | 6 |
| Programming | List Jobs page | 2 |
| Programming | Create job question | 2 |
| Programming | Create job approval | 2 |
| Programming | Create job correction | 3 |
| Programming | Job E-Mail functionality | 3 |
| Programming | Job-State-Handeling | 6 |
| Programming | Permission Management | 8 |
| Programming | Writing Unittests | 4 |

# Phase 5 (Budget / Market / Messages)

The fifth and last implementing phase contains following features:

* Ability to create a message for certain groups / users / market
* See the status of the different messages
* Notifications are sent according to the RE document
* Implementing the budget functionality in regard of the markets

Following pages are created:

* Message page
* List messages page
* Budget pages

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Design | Design Budget / Statistics Pages | 5 |
| Design | Design final version of search | 2 |
| Design | Design user profile page | 2 |
| CRUD | CRUD functionality budget | 6 |
| Programming | Connect Budget to Users & Markets | 4 |
| Programming | Implementing search (respecting permissions) | 3 |
| Programming | Implementing user profile page | 2 |

# Fixing DB schema / Refactoring

Refactoring is the process of cleaning the code. Making several optimizations in the code and the workflow. Extracting some common functionality in a core module and creating helper classes.

The DB schema will be finalized and should be stable for the release.

Not needed data will be erased.

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Database | Finalize Job / Task Schema | 3 |
| Database | Refactor current Database structure and code | 3 |

# Workflow / Beta – Alpha

**Alpha:**

During the alpha phase the application will be tested internally the.

The unittests will be updated to grant a stable and reliable application. Several bugs will be fixed.

Old not anymore needed data will be erased.

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Bugfixing | Bugfixing | 4 |
| Reserve | Reserve | 8 |
| Testing | Updating Unittests | 4 |

**Beta:**

Testversion for the customer. Feedback will be implemented and tests will be updated accordingly.

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Bugfixing 1 | Bugfixing | 4 |
| Testing 1 | Testing / Writing Unittests | 2 |
| Feedback 1 | React to feedback and make adjustments | 6 |
| Bugfixing 2 | Bugfixing | 4 |
| Testing 2 | Testing / Writing Unittests | 2 |
| Feedback 2 | React to feedback and make adjustments | 6 |

# Release / Go Live

Focus is on rolling out the application on the customer server. Implementing different notification mechanisms. Implementing a backup strategy.

**Release-Candidate**

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Bugfixing | Bugfixing | 2 |
| Testing | Testing / Writing Unittests | 0.5 |
| Feedback | React to feedback and make adjustments | 2 |

**Go-Live**

|  |  |  |
| --- | --- | --- |
| Subtype | Task | Est. duration in hrs |
| Communication | Communication | 3 |
| Database | Erasing not needed data | 2 |
| Backup | Setting up Backup | 4 |
| Backup | Setting up automatic reporting | 2 |

Additional Information

**Summary**

This chart shows the distribution of time required by the different phases.

Each task has a little time reserve for unpredictable difficulties and small changes.

Bigger changes from the user are not respected in the time.

|  |  |
| --- | --- |
| Phase – Type | Hours |
| Administrative | 94 |
| Preparation | 36 |
| Phase 1 | 35 |
| Phase 2 | 25 |
| Phase 3 | 25 |
| Phase 4 | 53 |
| Phase 5 | 24 |
| Workflow | 40 |
| Release & Go Live | 15.5 |
| Fixing DB Schema | 6 |
| Total | 353.5 |

Detailed time-table.

|  |  |
| --- | --- |
| Phase - Type | Time |
| Administrative | 94 |
| Preparation - Design | 20 |
| Phase 1 - Design | 8 |
| Preparation - Development | 16 |
| Phase 1 - Programming | 27 |
| Phase 2 - Design | 4 |
| Phase 2 - Programming | 21 |
| Phase 3 - Design | 3 |
| Phase 3 - Programming | 22 |
| Phase 4 - Designing | 9 |
| Fixing DB Schema | 6 |
| Phase 4 - Programming | 44 |
| Phase 5 - Design | 9 |
| Phase 5 - Programming | 15 |
| Workflow - Alpha | 16 |
| Workflow - Beta 1 | 12 |
| Workflow - Beta 2 | 12 |
| Release Candidate | 4.5 |
| Go Live | 11 |
| Total | 353.5 |

**Page-Tree:**

A list of required pages can be found in the interactive MindManager Application under following path:

<http://www.schmitz-dynamics.ch/wp-content/themes/quasi_single_app/assets/css/PageTree.html>

Several easy “create pages” (e.g. create new message) could be realized with modal’s instead of separated pages. (Depending on end customer preferences).

**Available Time:**

Following table shows my available time (about 70% percentage). This makes about 28 hours a week.

|  |  |  |
| --- | --- | --- |
| Month | Days | Hours |
| November | 15 | 120 |
| December | 11 | 88 |
| January | 17 | 136 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **November** | **December** | **January** |
| Administrative |  |  |  |
| Preparation |  |  |  |
| Phase 1 |  |  |  |
| Phase 2 |  |  |  |
| Phase 3 |  |  |  |
| Phase 4 |  |  |  |
| Phase 5 |  |  |  |
| Workflow |  |  |  |
| Release & Go Live |  |  |  |
| Fixing DB Schema |  |  |  |
|  |  |  |  |
| Partialy |  |  |  |
| Full |  |  |  |
|  |  |  |  |

**Costs**

The costs are calculated with a provisionally hourly rate of 130 CHF.

|  |  |  |
| --- | --- | --- |
| Sub-Types | Hours | Cost |
| Administrative | 94 | 12220 |
| Preparation | 36 | 4680 |
| Phase 1 | 35 | 4550 |
| Phase 2 | 25 | 3250 |
| Phase 3 | 25 | 3250 |
| Phase 4 | 53 | 6890 |
| Phase 5 | 24 | 3120 |
| Workflow | 40 | 5200 |
| Release & Go Live | 15.5 | 2015 |
| Fixing DB Schema | 6 | 780 |
| Total |  | 45955 |

With best regards



Daniel Schmitz

1. CRUD (Create, read, update, delete) [↑](#footnote-ref-1)