# Teilnehmer/innen des Teams:

|  |  |
| --- | --- |
| Klasse:  Bi20a | Team:  Aeschlimann, Kramer |

# Anforderungsdefinition (Meilenstein A)

|  |  |
| --- | --- |
| „Programm-Opener“ | |
| **Fachlicher Inhalt:**  (Allgemeine Beschreibung) | Nutzen: Mit dem Programm sollen zuvor festgelegte Programme geöffnet werden**Automation:**  Das Programm öffnet verschiedene Programme.  **Details:**   * Benutzerfreundlich * Gui erstellen   (Skizze / Mockup) |
| **MUSS**  **Kriterien:**  (Konkrete Features, die umzusetzen sind) | **Folgende Features sollen implementiert werden, um einen produktiven Ablauf sicherzustellen:**   * Mehrere zuvor Konfigurierte Programme öffnen * Zu öffnende Programme öffnen |

|  |  |
| --- | --- |
| **KANN**  **Kriterien:**  (Konkrete Features, die optional sind) | **Folgende Features können zusätzlich implementiert werden: (Varianten, Kreativität)**   * Dark Mode |

## Planung Meilensteine (LB1 / LB2)

|  |  |  |  |
| --- | --- | --- | --- |
| *MS* | *Tätigkeit / Abgabe* | *Soll-Datum* | *Ist-Datum* |
| A | ProjektstartTeam Bildung, Kollaborationsplattform, GitHub Repos \*, LehrerzugangWahl / Ausarbeitung der Anforderungsdefinition Abnahme Anforderungsdefinition durch Lehrperson | 28.09.21 | 28.09.21 |
| B | Teamaufgabe 1:Abgabe: Lösungsdesign  (Funktionsmodell / GUI / PAP / Storyboard) | 05.10.21 | 05.10.21 |
| B2 | Teamaufgabe 2: (Nur LB2)Abgabe: Testvorschrift und Testfälle | 09.11.21 | 09.11.21 |
| C | Einzelaufgabe 2 (LB1) / 3 (LB2):Abgabe Programmcode und DokumentationFachgespräch Projektabnahme | 09.11.21 | 09.11.21 |
| C2 | Einzelaufgabe 4: (Nur LB2)Abgabe: Ausgefüllter Systemtest | 09.11.21 | 09.11.21 |

\*) Öffentliche GitHub-URLs im Ablageordner auf dem BSCW ablegen! (pro Team)

Namenskonvention URL: **M122\_Klasse\_Thema\_Name\_Name**

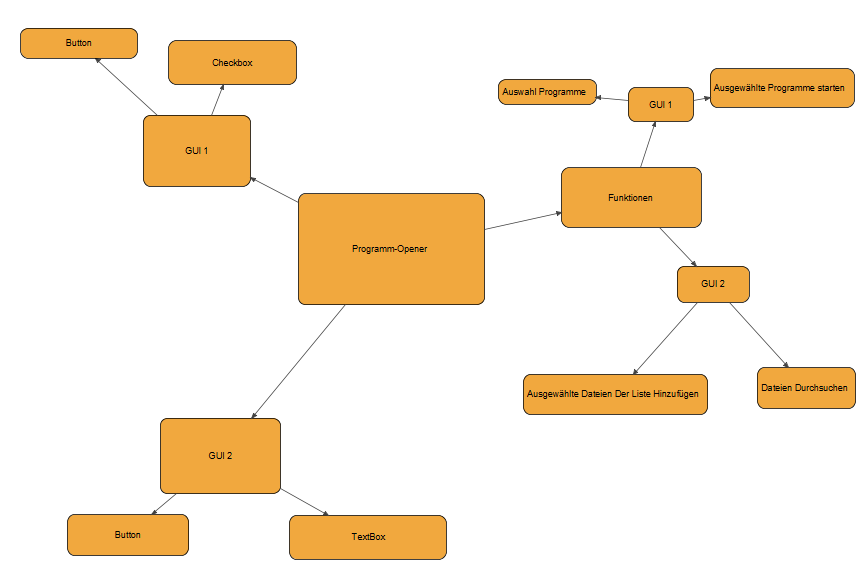
# Lösungsdesign (Meilenstein B: Teamaufgabe 1)

Anhand der Analyse wurde folgendes Lösungsdesign entworfen:

## Schematische Darstellung der Funktionalität, sog. Funktionsmodell

Im Folgenden ist die erwartete Funktionalität dargestellt und erklärt:

...



(**Funktionsmodell**: Skizze, Bild, Pictogramm, Mindmap, Blockdiagramm, UseCase (API) zur obigen Anforderungsdefinition **mit Legende**)

## Graphische Benutzer Schnittstelle (GUI) zur Konfiguration des Ablaufs

Durch diese Buttons kann man Automatisch seine Programme öffnen

Hier stehen die ausgewählten Programme drinnen. In dem kleinen Kästchen kann man sie ab- oder auswählen

Durch diese Buttons kommt man in das nächste GUI.

Das ist das Start GUI. Durch beide untere Buttons kommt man in das nächste GUI.

Ein Bild, das Text enthält.

Automatisch generierte BeschreibungDas ist das GUI für das hinzufügen der Programme

Feld, um den Pfad des Programmes hinzuzufügen oder mit dem Durchsuchen Button im Explorer suchen

Button um das ausgewählte Programm hinzuzufügen

## Ablauf der Automation

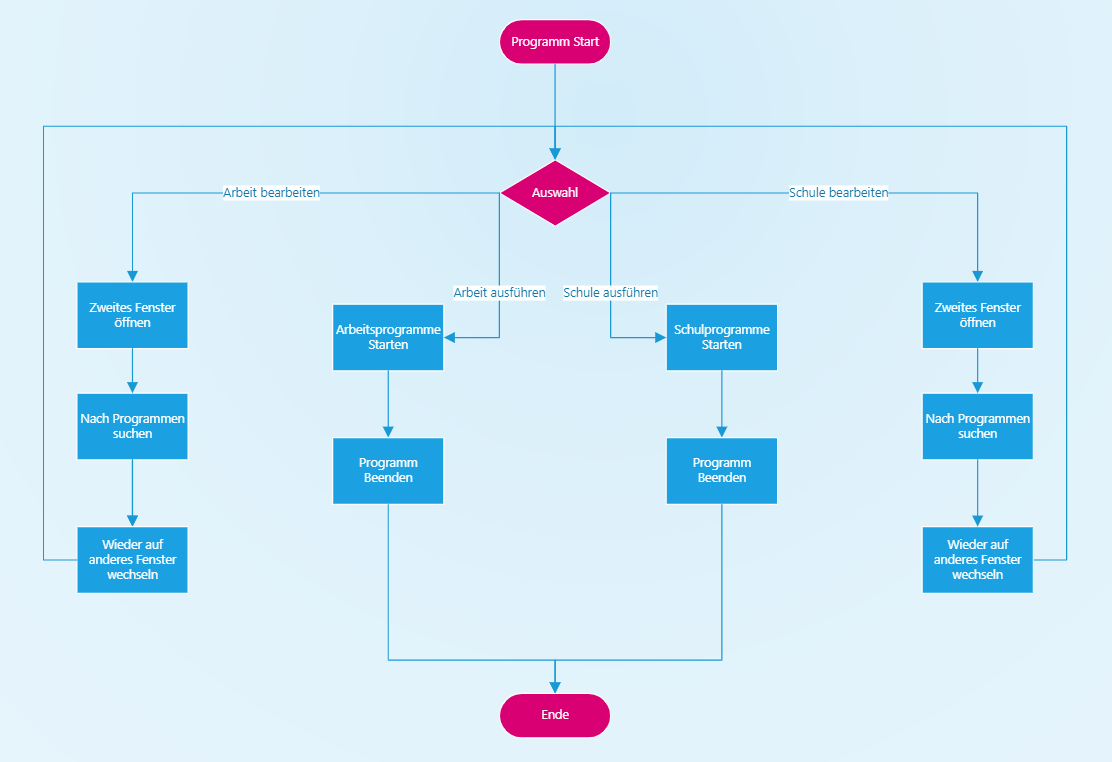
Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:

Siehe UML Aktivitätsdiagramm

Aus Administratorsicht ist folgender Ablauf des Programms zu erwarten:

Siehe UML Aktivitätsdiagramm

(Flussdiagramm (PAP / APIs: UML Aktivitätsdiagram) / Storyboard)



# Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

Testbeschrieb und vorbereitetes Testprotokoll siehe Dokument   
***M122\_LB2\_Testvorschrift\_MS-B2\_Aeschlimann\_Kramer.docx***

# Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument   
***M122\_LB2\_Testvorschrift\_MS-C2\_Aeschlimann\_Kramer\_Ausgefüllt.docx***

# Systemdokumentation (Meilenstein C: individuelle Aufgabe 3)

Die erstellten Projekt-WPS-Scripts sind hier abgelegt und für Entwickler dokumentiert:

Öffentliche GitHub-URLs im Ablageordner auf dem BSCW ablegen! (pro Team)

Namenskonvention URL: **M122\_Klasse\_Thema\_Name\_Name**

*à Ein* ***Branch*** *und separater Doku-Ordner pro Teammitglied erstellen*

## Umfang / Abgrenzung / Änderungen gegenüber Design

Keine Änderungen vorgenommen

## Funktionalität der Implementation.

$school\_opn\_btn\_OnClick Die ausgewählten Schulprogramme werden gestartet

$close\_btn\_Onclick Das Fenster wird geschlossen

$work\_opn\_btn\_OnClick Die ausgewählten Arbeitsprogramme werden gestartet

$work\_add\_btn\_OnClick Hier werden die Arbeitsprogramme hinzugefügt

$search\_btn\_OnClick Hier kann man die Programme im Explorer auswählen

$handler\_Mit\_Click Wenn ein Programm ausgewählt wurde, wird es der Liste hinzugefügt

$school\_add\_btn\_OnClick Hier werden die Schulprogramme hinzugefügt

# Betriebsdokumentation (Meilenstein C: individuelle Aufgabe 3)

Für Administrator und Benutzer wird folgende Anleitung ausgeliefert ...

## Installationsanleitung für Administratoren

Das .ps1 Skript wird gestartet.

## Bedienungsanleitung für Benutzer

Mit einem Klick auf den Button «Programme hinzufügen» kommt man auf ein weiteres Fenster, in dem man den Pfad manuell eingeben kann oder man sucht nach dem Programm, in dem man auf Durchsuchen klickt. Hat man ein Programm ausgewählt oder den Pfad angegeben kann man auf den Button «Programm hinufügen» drücken. Nun wird das Programm der Liste, welcher man das Programm hinzufügen wollte, hinzugefügt. Möchte man weitere Programme der Liste hinzufügen kann man einfach erneut auf den «Programme hinzufügen» Button klicken. Wenn man alle Programme hinzugefügt hat kann man die die man Starten möchte anhaken und auf den jeweiligen «Programme starten» Button drücken