# Teilnehmer/innen des Teams:

|  |  |
| --- | --- |
| Klasse:  AP18b | Team:  Bächli & Felix |

# Anforderungsdefinition (Meilenstein A)

|  |  |
| --- | --- |
| „Wallee“ | |
| **Fachlicher Inhalt:**  (Allgemeine Beschreibung) | Nutzen: Mit dem Programm sollen zufällige einfarbige Hintergründe generiert werden und gesetzt werden.**Automation:**  Es werden einfarbige Hintergründe generiert, abgespeichert und verwaltet (löschen, setzen...).  **Details:**   * Konfiguration: Bild Speicherungspfad, (siehe zusätzliche Konfigurationen in KANN Kriterien) * Integration: ermitteln von Screen-Grösse * Administration: Einstellungen sind via Terminal änderbar   **Erkenntnisse aus der Machbarkeitsabklärung in Node.js, Python, Perl:**  Alle von uns gesetzten MUSS und KANN Kriterien sind in Node.js, Python & Perl möglich |
| **MUSS**  **Kriterien:**  (Konkrete Features, die umzusetzen sind) | **Folgende Features sollen implementiert werden, um einen produktiven Ablauf sicherzustellen:**   * Ermitteln von Monitor-Grösse * Generieren von einfarbigen Wallpaper (.png) * Abspeichern von neuem Bild * Setzen von Wallpaper |

|  |  |
| --- | --- |
| **KANN**  **Kriterien:**  (Konkrete Features, die optional sind) | **Folgende Features können zusätzlich implementiert werden: (Varianten, Kreativität)**   * Löschen von alten Bildern (+ max alte Bilder in config änderbar) * Script wird bei Start von PC gestartet (in config änderbar) * «Wallee –config» Bsp. für Einstellungen * Auswahl von normalen Bildern in explizitem Ordner (in config änderbar) |

## 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 |  |  |
| B | Teamaufgabe 1:Abgabe: Lösungsdesign  (Funktionsmodell / GUI / PAP / Storyboard) |  |  |
| B2 | Teamaufgabe 2: (Nur LB2)Abgabe: Testvorschrift und Testfälle |  |  |
| C | Einzelaufgabe 2 (LB1) / 3 (LB2):Abgabe Programmcode und DokumentationFachgespräch Projektabnahme |  |  |
| C2 | Einzelaufgabe 4: (Nur LB2)Abgabe: Ausgefüllter Systemtest |  |  |

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

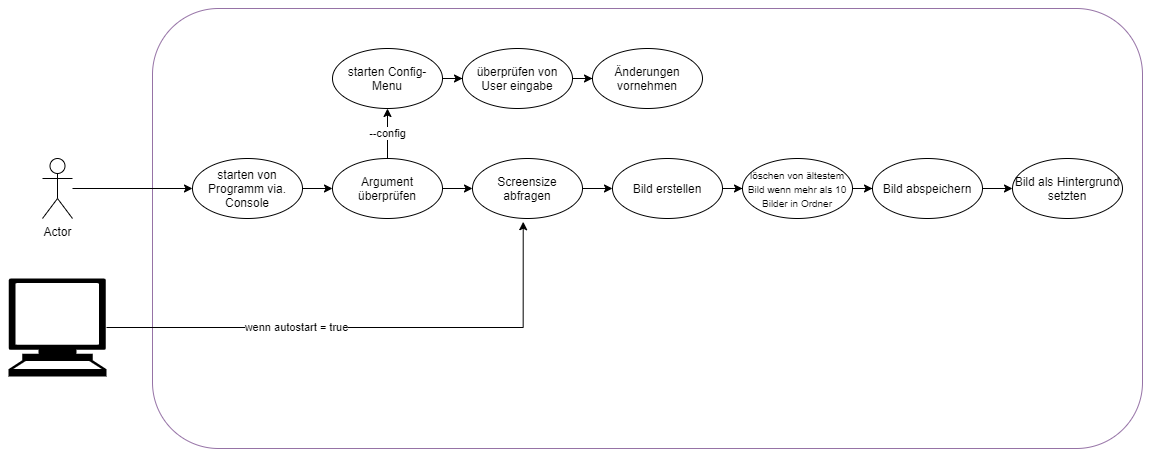
Namenskonvention URL: **M122\_Klasse\_Thema\_Name\_Name**

# 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:

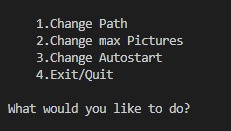


(**Funktionsmodell**: Skizze, Bild, Pictogramm, Mindmap, Blockdiagramm, UseCase (API) zur obigen Anforderungsdefinition **mit Legende**)

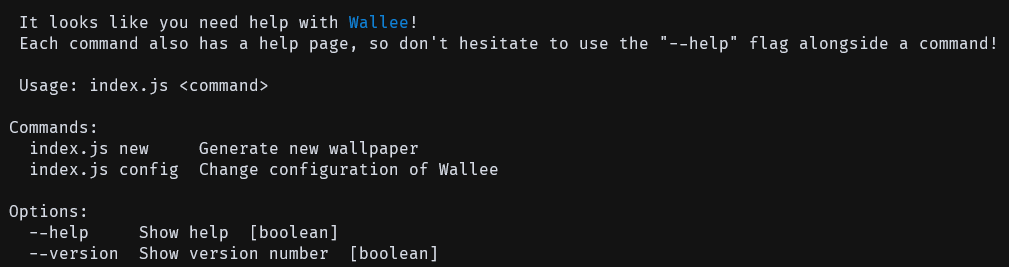
## Graphische Benutzer Schnittstelle (TUI) zur Konfiguration des Ablaufs

Das zu erwartende GUI ist hier dargestellt:

Lara:



Soryn:

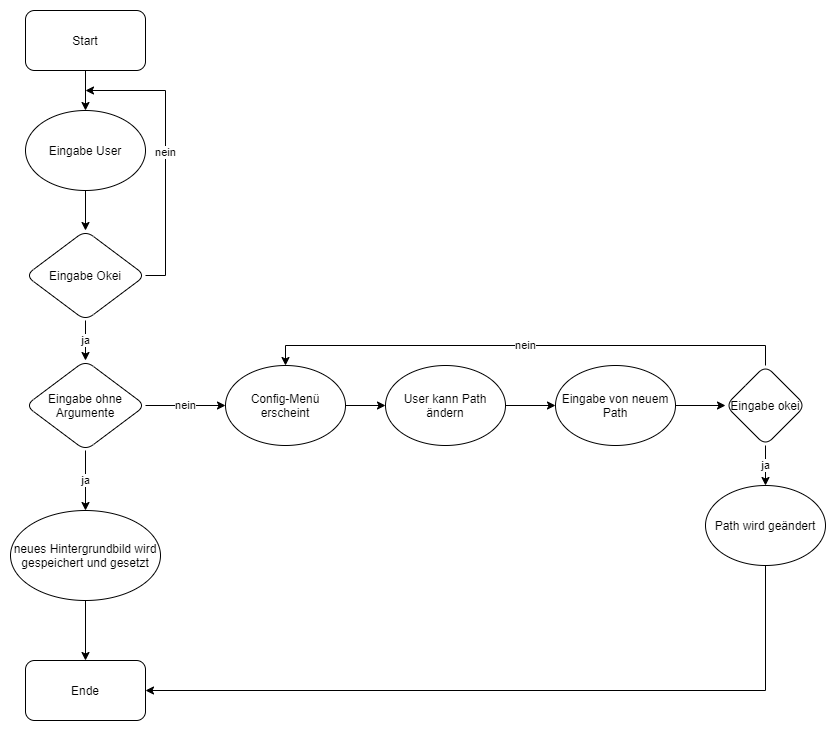


*(index.js wird in Produktion zu wallee)*

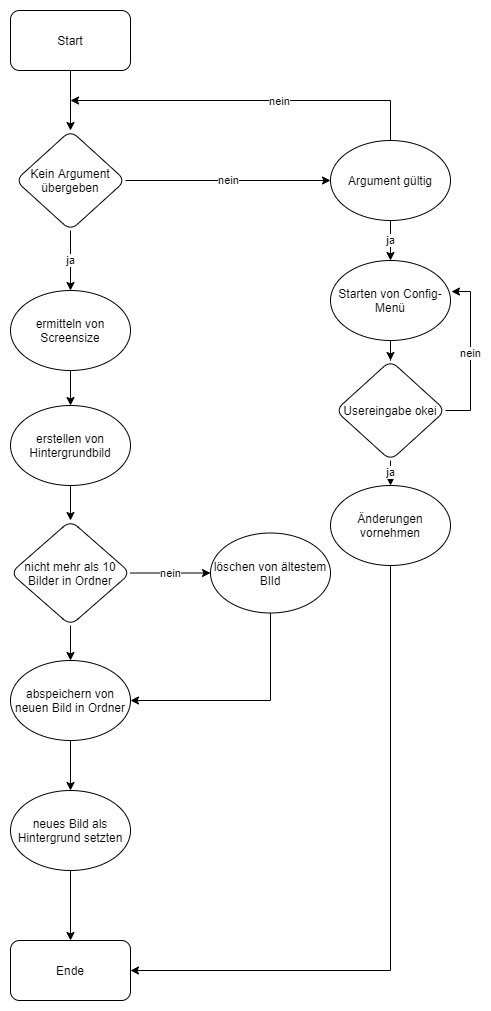
(GUI Formulare mit Beschreibung)

## Ablauf der Automation

Aus Benutzersicht ist folgender Ablauf des Programms zu erwarten:



Aus Administratorsicht ist folgender Ablauf des Programms zu erwarten:



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

# Testvorschrift (LB2 Meilenstein B2: Teamaufgabe 2)

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

# Testprotokoll (LB2 Meilenstein C2: individuelle Aufgabe 4)

Ausgefülltes Testprotokoll siehe Dokument   
***M122\_LB2\_Testvorschrift\_MS-C2\_Namen.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)

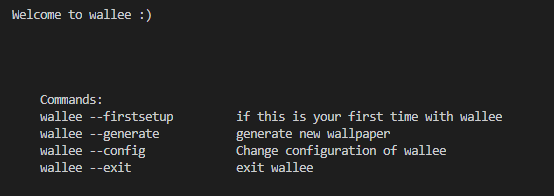
Namenskonvention URL: **M122\_Klasse\_Thema\_Name\_Name**

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

## Umfang / Abgrenzung / Änderungen gegenüber Design

Aufgrund unten beschriebener Umstände sind Anpassungen des ursprünglichen Lösungsdesigns gemacht worden:

*Zusätzlich zu dem oben definierten TUI wurde noch ein Hauptmenü erstellt, indem man wie geplant mit dem Command ‘wallee --config’ in das Config-Menü kommt (siehe Screnneshot oben). Zudem gibt es noch die Commands ‘wallee --firstsetup’ in dem man den Speicherort der Bilder setzen kann, mit ‘wallee --generate’ kann man ein neues Hintergrundbild generieren und setzen. Das Menü kann man mit dem Command ‘wallee --exit’ schliessen.*



Umstände / Anpassungen / Veränderungen

## Funktionalität der Implementation.

Zusätzlich zu der Inline-Dokumentation sind hier folgende Funktionen / Eventhandler detailliert beschrieben:

...

Ausführliche Beschreibung der internen Funktionen (Eventhandler wie z.B: $btn\_ok\_Click), der Parameter und der Rückgabewerte

**Struktogramm** (BET / SYS) oder **UML Aktivitätsdiagramm** (API)

# Betriebsdokumentation (Meilenstein C: individuelle Aufgabe 3)

Für Administrator und Benutzer wird folgende Anleitung ausgeliefert ...

## Installationsanleitung für Administratoren

Das Programm ist folgendermassen zu installieren und konfiguriert …

*Angegeben in README.md im Repository als ‘dev - instructions’*

## Bedienungsanleitung für Benutzer

Das Programm ist folgendermassen zu bedienen ...  
*Angegeben in README.md im Repository als ‘user - instructions’*