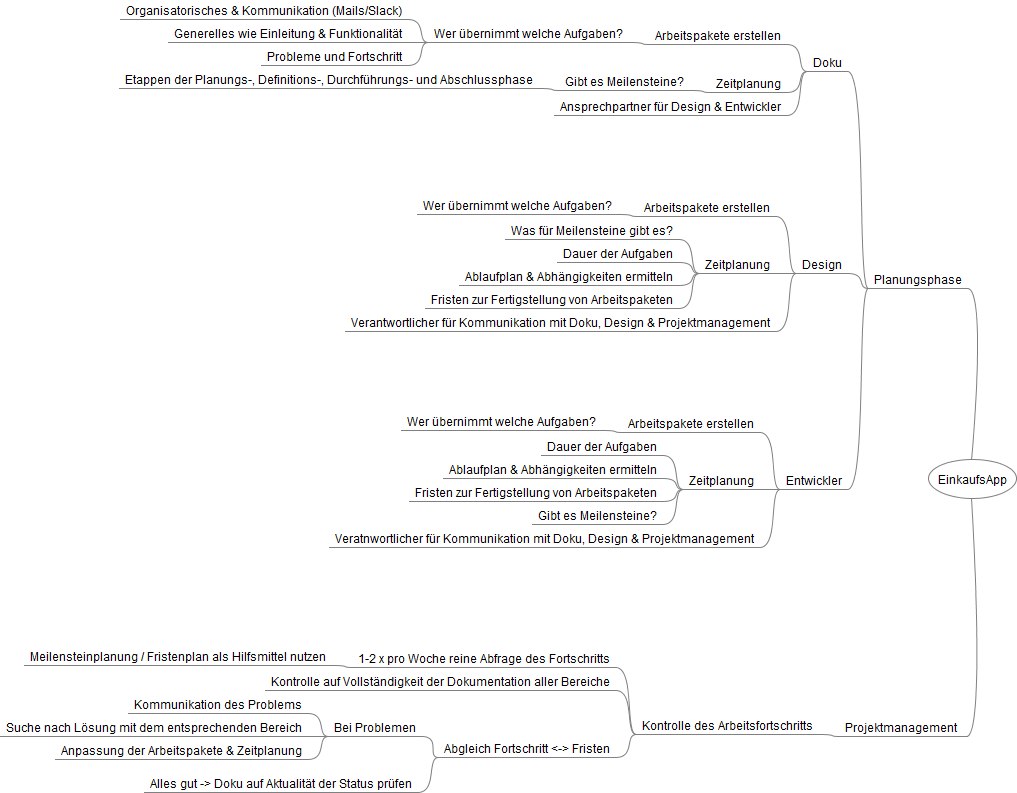
**1.4.5. Arbeitsplanung**

Zu Beginn der Projektorganisation wurde von der Dokumentation ein grober Plan bereitgestellt, der eine Einteilung der Teams in organisatorische Einheiten festschreibt und einen Rahmen für die Planung der Aufgaben beziehungsweise Arbeitspakete vorgibt. Damit wurde ein organisatorisches Grundgerüst geschaffen, das allen Gruppen als Orientierung dient und gleichzeitig zur eigenständigen Organisation und Erstellung sowie Bearbeitung der Arbeitspakete motiviert:

Mindmap: Organisation



**Meilensteine**:

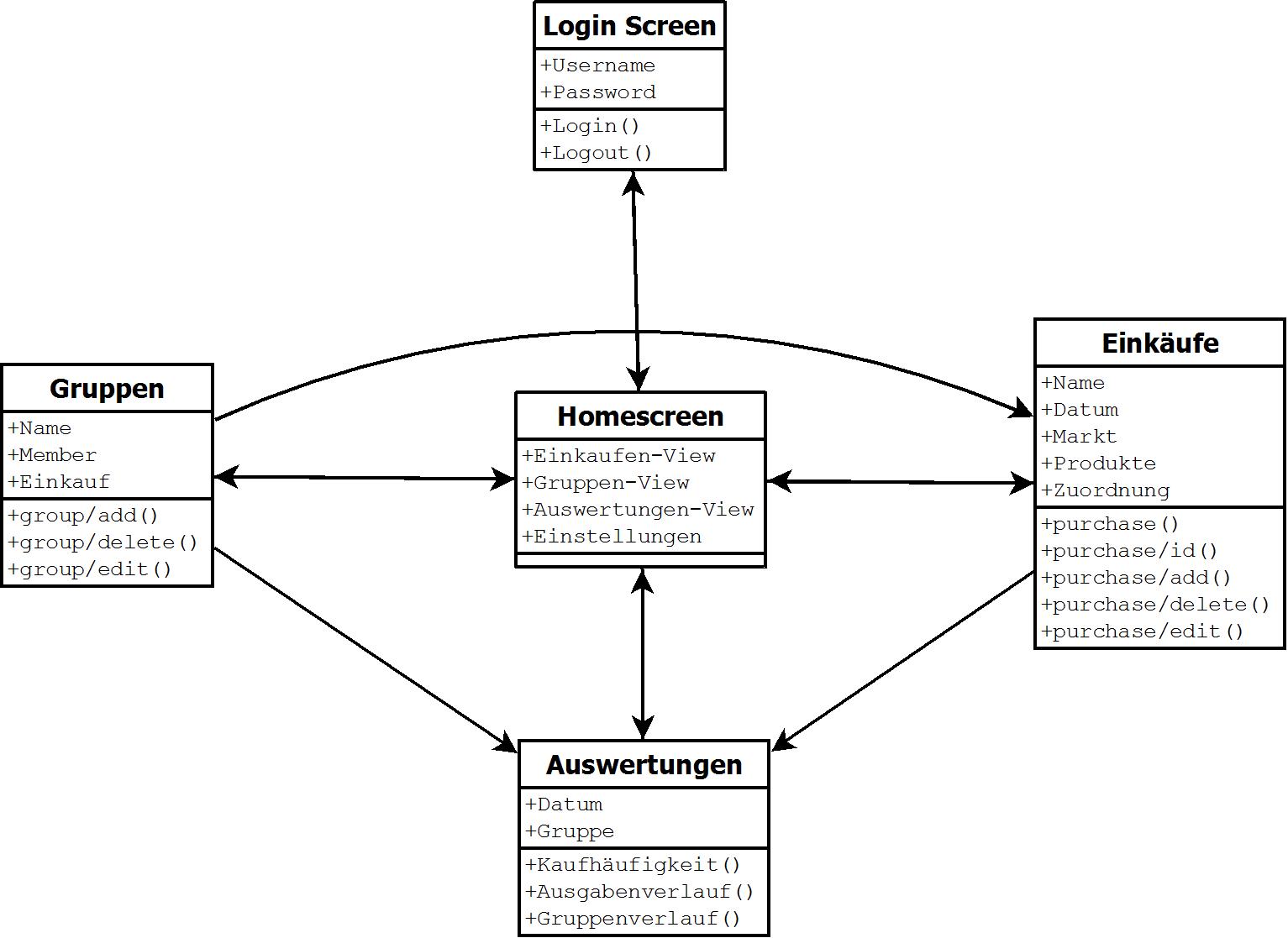
Da die Meilensteinplanung hat als Planungsgrundlage die Verfügbarkeiten für persönliche Treffen der einzelnen Teams genutzt. Die Fertigstellung der Arbeitspakete wurde so koordiniert, dass sich die Teams für die Mehrheit der Arbeitspakete ein einziges Zieldatum ausgewählt haben, zu dem alle Mitglieder präsent waren, um die Ergebnisse gemeinsam zu verifizieren.

|  |  |  |
| --- | --- | --- |
| **Datum** | **Gruppe** | **Meilenstein** |
| 23.10.2015 | des | Fertigstellung Login |
| 30.10.2015 | dev | Fertigstellung Login |
| 06.11.2015 | dev | Fertigstellung Menü |
| 06.11.2015 | des | Fertigstellung Einkaufsmodul |
| 13.11.2015 | doku | Handbuch Entwurf erstellen basiert auf Development Status |
| 13.11.2015 | dev | Fertigstellung Einkaufsmodul |
| 13.11.2015 | des | Fertigstellung Auswertungsmodul |
| 20.11.2015 | doku | Fertigstellung Struktur der Doku und 1. Version des Handbuchs |
| 20.11.2015 | dev | Fertigstellung Auswertungsmodul |
| 20.11.2015 | des | Fertigstellung Gruppenverwaltungsmodul |
| 20.11.2015 | alle | Evaluation aller Dokumente, die erstellt werden müssen (UMLs…) |
| 27.11.2015 | dev | Fertigstellung Gruppenverwaltungsmodul |
| 04.12.2015 | doku | Anfang Schreiben der finalen Doku und des Handbuchs |
| 04.12.2015 | dev | App Fertigstellung - Pen tests |
| 04.12.2015 | alle | AZ Woche - Dokumentation, Fehlerbehebung, usw |
| 11.12.2015 | alle | App finale Fertigstellung |
| 18.12.2015 | alle | Dokufertigstellung und Handbuch |

**Diagramme:**

Entsprechend den Projektanforderungen wurden fünf verschiedene Diagramm-Typen erstellt. Jeder dieser fünf Diagramm-Typen dient der Strukturierung der Systemarchitektur oder der Prozessbeschreibung.

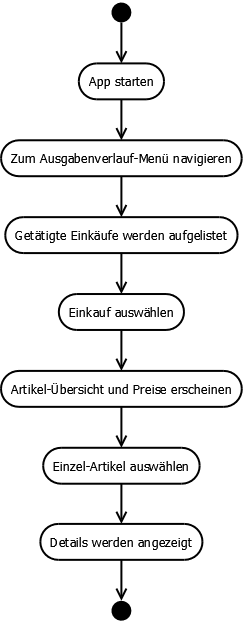
**Klassen-Diagramm:**



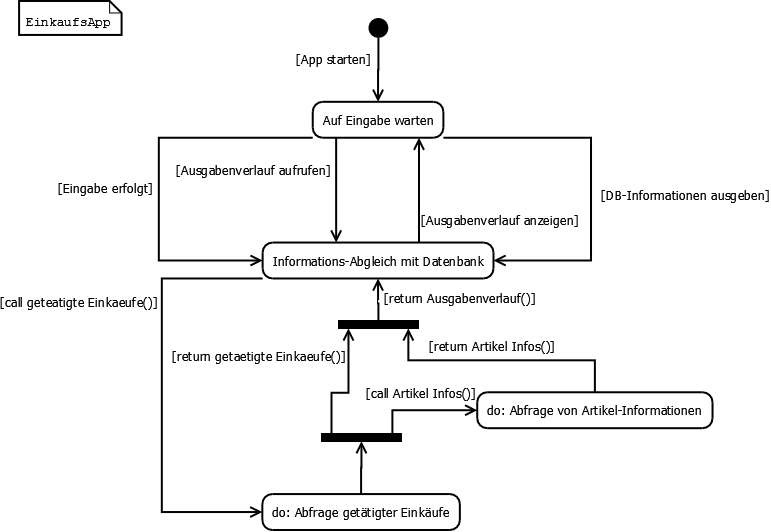
**Aktivitäts-Diagramm „Einkauf einlesen“:**



**Aktivitäts-Diagramm „Ausgabenverlauf anzeigen“:**



**Zustands-Diagramm „Ausgabenverlauf anzeigen“:**



**Sequenz-Diagramm:**

„Ausgabenverlauf anzeigen“

****

**Sequenz-Diagramm:**

„Einkauf einlesen“

****

**Anwendungsfall-Diagramm:**

„Einkauf einlesen“

|  |  |
| --- | --- |
| **use case** | Einkauf einlesen |
| **actors** |  |
|  | Benutzer |
| **precondition** |  |
|  | Artikel hat erkennbaren Barcode, Rahmenbedingungen das die App ausgeführt werden kann sind gegeben ( Akku voll, genug Rechenleistung etc) , App ist gestartet, Kamera des Mobiltelefons ist funktionstüchtig. |
| **main flow** | Artikel bekannt |
|  | Der Nutzer startet die App auf seinem Smartphone. Er navigiert beim Start des Einkaufes in der App durch das Menü und setzt alle Einstellungen die er haben möchte. Er startet den Einkauf, wählt den Supermarkt im Menü aus, in dem er sich befindet, er nimmt den gewünschten Artikel den er einlesen (einkaufen) möchte aus dem Regal im Supermarkt und richtet den Artikel so aus, dass er den Barcode mit dem integrierten Barcodescanner scannen kann. Wenn dieser erfolgreich gescannt ist, sieht er auf dem Screen alle erfassten Informationen. Daraufhin kann er die Menge auswählen ,die er einkaufen möchte und anklicken, ob dieser Artikel für ihn oder z. B. jemand aus seiner Gruppenverwaltung bestimmt ist und ob der Preis oder Sonderpreis gerade gültig ist. Nun kann er den nächsten Artikel Scannen |
| **alternitive flow** | Artikel nicht bekannt |
|  | Der Nutzer startet die App auf seinem Smartphone. Anschließend navigiert er beim Start des Einkaufes in der App durch das Menü und konfiguriert alle Einstellungen nach seinen Wünschen. Wenn er den Einkauf startet, wählt den Supermarkt über das Menü aus, in dem er sich befindet. Im Markt nimmt er den gewünschten Artikel den er einlesen (einkaufen) möchte aus dem Regal und richtet den Artikel so aus, dass er den Barcode mit dem integrierten Barcodescanner scannen kann. Wenn der Artikel nicht im System hinterlegt ist, kann er diesen hinzufügen. Hierfür gibt er Preis, Inhaltsmenge, die Einheit der Inhaltsmenge und den Titel an. Nun kann er die Menge auswählen die er einkaufen möchte und anklicken, ob dieser Artikel für ihn oder z. B. jemand aus seiner Gruppenverwaltung bestimmt ist und ob der Preis oder Sonderpreis gerade gültig ist. Nun kann er den nächsten Artikel Scannen |
| **postcondition** |  |
|  | Der Artikel wurde erfolgreich gescannt. Die Daten im System wurden auf Aktuellität durch den User kontrolliert oder aktualisiert. Der Artikel ist nun im Einkaufskorb aufgenommen und für wen dieser eingekauft wurde. |
| **exceptional flow** | Ausnahme 1 |
|  | Der Artikel hat keinen Barcode |
| **exceptional flow 2** | Ausnahme 2 |
|  |  |
| **post condition** |  |
|  | Der Artikel kann nicht im System erfasst werden, da dieser keinen Barcode hat. |
| **end** | Einkauf einlesen |

**Anwendungsfall-Diagramm:**

„Ausgabenverlauf anzeigen“

|  |  |
| --- | --- |
| **use case** | Ausgabenverlauf anzeigen |
| **actors** |  |
|  | Benutzer |
| **precondition** |  |
|  | Rahmenbedingungen, dass die App ausgeführt werden kann sind gegeben ( Akku voll, genug Rechenleistung etc) , App ist gestartet, es sind bereits Einkäufe getätigt worden |
| **main flow** | Ausgaben bekannt |
|  | Der Nutzer startet die App auf seinem Smartphone. Er navigiert beim Start des Einkaufes in der App durch das Menü auf Ausgabenverlauf anzeigen. Er bekommt eine Auflistung aller letzten getätigten Einkäufe mit Datum, Uhrzeit, Supermarkt und dem eigens vergebenen Titel dieses Einkaufens. Nun kann er einen gewünschten Einkauf auswählen und bekommt eine Übersicht von allen Artikeln mit Preis. Für Details der Artikel muss er den Artikel auswählen. |
| **alternitive flow** |  |
|  |  |
| **postcondition** |  |
|  | Der Ausgabenverlauf wird erfolgreich mit allen Details angezeigt, sodass der Nutzer seine Ausgaben stets im Blick hat. |
| **exceptional flow** | Ausnahme 1 |
|  | Es ist noch kein Einkauf getätigt worden |
| **exceptional flow 2** | Ausnahme 2 |
|  |  |
| **post condition** |  |
|  | Der Ausgabenverlauf ist leer. |
| **end** | Ausgabenverlauf anzeigen |

**1.4.6. Agiles Projektmanagement**

Nachdem ein grober organisatorischer Rahmen für das Projekt von der Gruppe Dokumentation vorgegeben wurde, haben die einzelnen Gruppen intern mittels agiler Projektmanagement-Methoden ihre Arbeitspakete und Ablaufpläne festgelegt.

Mittels Telefonkonferenzen bei der Gruppe Dokumentation wurden Arbeitspakete erstellt und jeder hat sich nach dem Pull-Prinzip – bekannt aus der Projektmanagement-Methode Kanban – seine Arbeitspakete abgeholt und eine Bearbeitungsfrist definiert.

In der Gruppe der Entwickler wurde unter Zuhilfenahme des Tools „Trello“ die Planung und Durchführung der Arbeitspakete durchgeführt. Trello ist ein Web-Dienst, der ein Board bietet, um Arbeitspakete gemäß agiler Projektmanagement-Methoden zu bearbeiten und Arbeitsfortschritte transparent darzustellen.

Die Gruppe Design hat ihre Arbeitspakete auf Basis eines Ablaufplanes auf die verschiedenen Mitglieder verteilt. Es wurden die Phasen Prototyp-Entwurf, Prototyp-Review, Prototyp-Modifikationen und Prototyp-Test und Prototyp-Abnahme durchlaufen.

**1.5. Sicherheit**

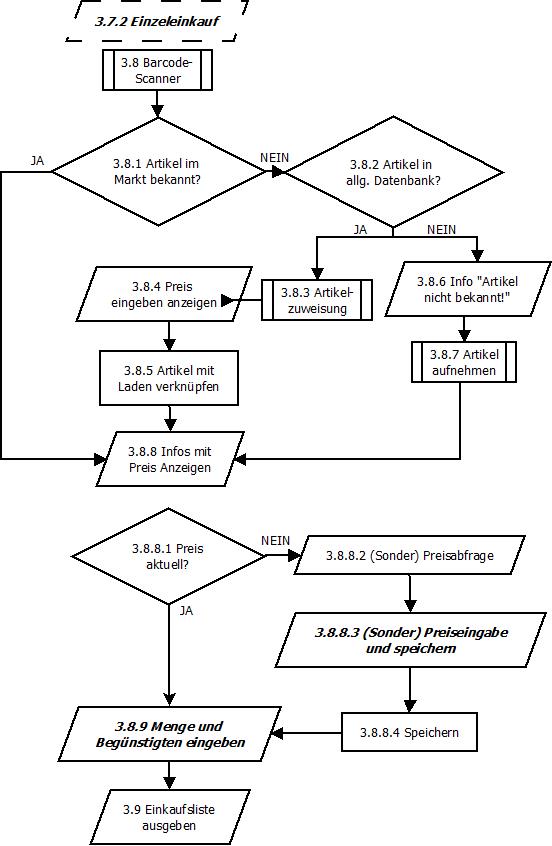
Zwecks Sicherheit wurde bei der Entwicklung der EinkaufsApp von Anfang an auf sicherheitsorientierte Programmierung geachtet. Beispielsweise Passwörter werden nicht im Klartext gespeichert und Variablen nicht als Public deklariert. Zudem hat sich das Entwicklungsteam für eine zentrale Datenbank entschieden, die Daten nicht lokal speichert. Somit sollen Daten nicht vollständig aus dem Endgerät ausgelesen werden können.

Einige konkrete Sicherheitsvorkehrungen lauten:

1. Eine Session läuft nach 10 Minuten ohne Request an den Server ab.
2. Jedes Passwort wird als Einweg-Hash in der DB gespeichert.
3. Es wird eine Authentifizierung durchgeführt, das heißt ein Sicherstellen, dass die beteiligten Parteien auch tatsächlich die sind, die sie vorgeben zu sein.
4. Data Security: Eine weitere wichtige Technik zur Sicherheit von Webanwendungen spielt das SSL-Protokoll (Secure Socket Layer). Bei Verwendung dieser Protokolle wird vor der eigentlichen HTTP-Kommunikation ein geschützter Kanal aufgebaut,   
   sodass die Nutzerdaten für Dritte nicht zugänglich sind. Die Nutzerdaten werden in einem normalen Web-Formular eingetragen (Login-Screen) und mittels POST-Request an den Server gesendet. Da der TCP-Kanal abgesichert ist, haben Dritte keinen Zugriff auf die   
   Nutzdaten innerhalb des POST-Requests, d.h. Nutzername und Passwort werden gesichert übertragen.

**2.4. Einkaufsverwaltung**

**Design (Ausschnitt aus einem Einzeleinkauf)**

****

Das Design Team hat ein Flussdiagramm zur Einkaufsverwaltung erstellt, das den Ausbau der Systemarchitektur berücksichtigt und als Orientierung für die Entwickler gilt. Zwecks Übersichtlichkeit ist nur ein Teil des Flussdiagramms enthalten (Anm. d. Autors: Das vollständige Flussdiagramm befindet sich im Anhang).

Gleichzeitig wird ein Prototyp in Proto.io entwickelt, mit der Zielsetzung, die Hauptfunktionalität der Anwendung abzubilden. Das Tool gibt bereits im frühen Projektstadium, vor der eigentlichen Entwicklung, ein Gefühl für das Endprodukt.

Insbesondere bei der Umsetzung der Designvorgaben durch die Entwicklung ist es eine große Herausforderung sich untereinander abzustimmen und speziell die Umsetzbarkeit von Designvorgaben zu prüfen.

**Entwicklung**

Da das Projekt enorm zeitkritisch war sind Design und Implementation stark parallelisiert abgelaufen. Um innerhalb von zwei Monaten ein lauffähiges und qualitätsgesichertes Software Produkt zu erstellen bedarf es eines überschaubaren und für alle Mitglieder transparenten Projektes.

Im Verlauf des Projektes wurde eine zusätzliche Organisationseinheit zwischen dem Gesamtprojekt und den einzelnen Entwicklern eingefügt, um einen besseren Überblick zu gewährleisten und schneller Entscheidungen treffen zu können.

Durch eine klare Aufgabentrennung, die anhand der einzelnen Module der EinkaufsApp abgegrenzt wurde, wurden die verschiedenen Kenntnisse der Entwickler berücksichtigt.

Zum Schluss mussten schließlich die Schnittstellen angepasst und die Kompatibilität der einzelnen Module geprüft werden.