Functioneel ontwerp

**Project P4P: Koffiekaart**

**Anne Pieter Boonstra**

**Jaimy Beima**

**Jesse Piebenga**

**Robert Slomp**

**Wijbe jan Zwager**

**Groep 28**

**Begeleider: de heer van der Meulen**

Tutor: Greta Schaafsma

**Leeuwarden**

**05-05-16**

**NHL Leeuwarden**

## Probleemstelling

Op dit moment is er een systeem op de NHL om het aantal koppen koffie dat je hebt aangeschaft te meten. Als je 10 koppen koffie hebt aangeschaft krijg je de elfde gratis. Dat doen de kassamedewerkers doormiddel van een stempelkaart die de klanten bij zich moeten houden. Alleen veel mensen vergeten hem of raken deze kwijt. Het percentage van de klanten die er gebruik van maakt is daardoor lager dat het kan zijn.

## Doelstelling

Een prototype van de kaartlezer met pasjes maken die het ouderwetse stempelkaart systeem kan vervangen waarbij je met een schoolpasje via RF-ID de klant met 1-stap de schoolpas tegen de kaartlezer kan leggen. De kassa moet automatisch kijken hoeveel koppen koffie er zijn aangeschaft en het corresponderende aantal bijschrijven op de schoolpas.

## Programma van eisen

Na met onze opdrachtgever gesproken te hebben, hebben wij een programma van eisen opgesteld. Zo konden wij makkelijk overzien wat wij wel een niet hoeven toe te voegen aan ons project. Wij werken samen met drie tweedejaars Bimmers, die al een deel van het werk hebben verricht. De eisen zijn vrij eenvoudig en makkelijk te overzien. Veel wensen waren er niet. Toch mogen wij zelf nog dingen bij het project bedenken als onze opdrachtgever daar mee akkoord gaat.

|  |  |
| --- | --- |
| **Eisen** | **Wensen** |
| * Een systeem dat een studentenkaart kan lezen. * Code die bij de kaart punten kan bijschrijven of resetten. * Code die deze data opslaat in een database. | * Een interface voor de kassamedewerkers. * Een database op een server om zo overal de juiste data te delen. |

## Randvoorwaarden

**Tijd**: het project moet voor 17-06-2016 ingeleverd worden

**Geld**: we hebben geen budget voor het project.

**Kwaliteit**:

De eisen die we gekregen hebben moeten 100% werkend zijn voordat we het inleveren, ons programma moet het minst mogelijk aantal bugs hebben.

**Overig**:

Het project word beheerd met GIT

## Planning

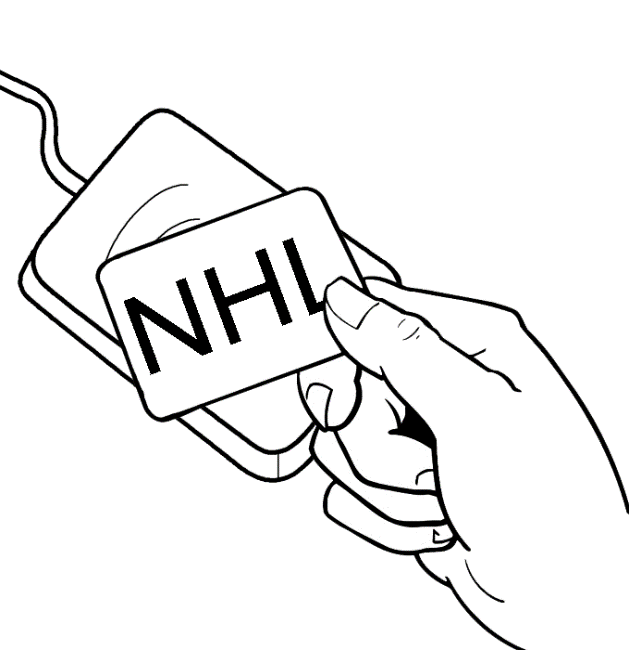
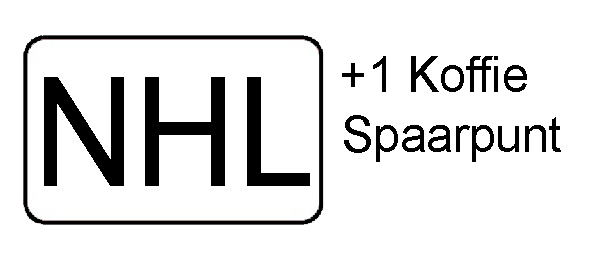
Als groep hebben wij een mooi strak schema. Wij komen minstens twee keer per week bij elkaar om de stand van zaken te bespreken en om te kijken wat wij in de toekomst kunnen doen.

Wij spreken 1 keer per week af met onze tutor (Greta) om te kijken hoe ver we zijn en of de samenwerking tussen de groep goed verloopt, ook komt Greta met goede tips die wij gebruiken in onze groep.

Ook spreken wij 1 keer in de twee weken af met onze opdrachtgever om hem een update te geven en te laten zien wat wij tot nu toe hebben gemaakt, ook zullen er waarschijnlijk nieuwe eisen en wensen bijkomen die wij dan proberen toe te passen in ons project.

|  |  |
| --- | --- |
| 10-05 | Gesprek met tutor.  Stand van zaken bespreken en verder werken. |
| 11-05 | Gesprek met opdrachtgever.  Nieuwe eisen en wensen verwerken in plan van aanpak.  Planning met tweedejaars maken voor gebruik dummy reader. |
| 12-05 | Stand van zaken bespreken en verder werken. |
| 17-05 | Gesprek met tutor.  Stand van zaken bespreken en verder werken. |
| 19-05 | Stand van zaken bespreken en verder werken. |
| 24-05 | Gesprek met tutor.  Stand van zaken bespreken en verder werken. |
| 25-05 | Gesprek met opdrachtgever.  Nieuwe eisen en wensen verwerken in plan van aanpak. |
| 26-05 | Stand van zaken bespreken en verder werken. |
| 31-05 | Gesprek met tutor.  Stand van zaken bespreken en verder werken. |
| 2-06 | Stand van zaken bespreken en verder werken. |
| 9-06 | Gesprek met Tutor.  Project afronden. |
| 11-06 | Gesprek met opdrachtgever.  Puntjes op de i zetten. |
| 16-06 | Project nog een keer overkijken.  Kijken of wij tevreden zijn met het eindproduct en bespreken hoe het samenwerken ging.  Opdracht inleveren. |

## Visuele representatie



Punten worden automatisch toegekend aan de klant

Klant pakt NHL-pas

Klant legt NHL-pas op de scanner