Preface

Our intentions with this report is to give the reader insight into the processes we have used doing the project. As well as the development of the system from brief, user stories about our requirements and some diagram showing a bit about our analysis of the program, and hope fully have a draft of the program for the customer that is to his satisfaction.

We have each been involved with the entire development process, but responsibility for the individual areas has been split across the team :

Programming Servlet:   
 Cornel Lungeanu.

Programming Domain:  
 Kristian Hansen Buck.

Programming Technical:  
 Asger Jessen.

SQL query:   
 Asger Jessen.

Diagrams, User stories:  
 the Team.

// to be deleted

This report handles the group’s work on a given project. This project is about developing a Bike tilting game planner system, which is a distributed on an IT-System to support signing up, Lane creations, ongoing results registration and the communication of results

Indholdsfortegnelse

[Introduction 3](#_Toc437850912)

[Problem Analysis 4](#_Toc437850913)

[Subjects Programming (SDP) and Technology (CDS) 5](#_Toc437850914)

[Introduction to SDP/CDS 5](#_Toc437850915)

[Main Section 5](#_Toc437850916)

[Conclution for SDP/CDS 5](#_Toc437850917)

[System Development (SDM) 6](#_Toc437850918)

[Scrum introduction 6](#_Toc437850919)

[Main Section 7](#_Toc437850920)

[Sprints 7](#_Toc437850921)

[Kanban board 8](#_Toc437850922)

[Program Framework 8](#_Toc437850923)

[Program Screen Shots 8](#_Toc437850924)

[Signing up 8](#_Toc437850925)

[Adding Score 8](#_Toc437850926)

[Watching score Live 8](#_Toc437850927)

[Apendix 8](#_Toc437850928)

# Introduction

This Assignment is about creating a WEB-solution for a Bike tilting game. This problem should be replacing the current “system” which is done manually by pen and paper. The program is expected to handle all kinds of Bike tilting tournaments, with different kind of age groups and maybe later even being able to handle normal tilting, with op to the max of lanes without effecting the core of the program

The system should be able to handle the signing up for the bike tilting, managing the Lane making in the different age groups, live updating results for the audience, access control is part of the system various volunteers that must log in to the system

Since this should be a web distributed program so the volunteers are able to handle their job with their own mobile devices. This implies that the servlet is needed where the volunteers/parents can connect and operate/observe the tilting.

# Problem Analysis

An app is what the Bike tilting needs, but as we cannot make apps yet, we decided to make a web-application so that you are still able to use your tablet/mobile device instead of a laptop.   
The web-application should be able to create a Bike tilting tournament with the different age groups,   
while having that made it is also important that each age group can have different kind of rules, on how many runs they have in each age group.

When the Tournament is created, the signup period is 1½ month before. Another problem is that if there is a last participant at the last second that signs up must be put in the pool over the lanes and age groups, and be added to a lane and giving a shirt color and number. If there is not enough participants to fill out the lanes they will just remove some lanes or just fill the lanes with fewer participants.

The score needs to be handles by the volunteers, and when they all had the last run in the end the result should be shown, those with the most points should go on to the next stage but not till all participants had their run in the different age groups.

# Subjects Programming (SDP) and Technology (CDS)

## Introduction to SDP/CDS

## Main Section

## Conclution for SDP/CDS

# System Development (SDM)

## Scrum introduction

Scrum is an iterative and incremental agile software development methodology for managing a product development as the “Bike tilting”. The method is very flexible, meaning that even when we are working on the product the customer is able to change his/her mind about that they want and need, (often called “requirements volatility”).  
The team, works as a group where they want to reach a common goal. This enables the team to self-organize by encouraging physical co-location or close online collaboration of all team members, as well as a daily face-to-face communication among all team members and disciplines in the project.

Each project consists of various amount of groups. Each group has a development team that consists of 3-9 individuals who do the actual work. The work consists of. Analysis, design, develop, test, technical communication, documents, etc. The development teams are cross functional, with all of the skills as a team necessary to create the product increment. Each increment is a part of the bigger product, where they divided it up in smaller parts. Each team gets a part of the product they need to develop and then when it is completed all teams combine the parts they made and test the product to see if it works.

Scrum Master. Is the person that is accountable for removing impediments to the ability of the team to deliver the product goal. A Scrum master is not a traditional team leader or a project manager, but acts as a buffer between the team and any distracting influences, also makes sure that the developer’s uses scrum as intended. He/she also helps ensure the team follows the agreed scrum processes, of then facilitates key sessions, and encourages the team to improve.  
The Scrum Master role referred as a team facilitator or servant-leader to reinforce dual perspectives.

Work flow, is divided into sprints, the sprint is a timeboxed effort; witch means that it is restricted to a specific duration. The duration is fixed in advance for each sprint and is normally between one ween and one month, with two weeks being the most common. Each sprint starts with a sprint-planning event that aims to define a sprint backlog, identify the work for the sprint, and make an estimated commitment for the sprint goal. Each sprint ends with a sprint review where they look back to see what goal they did achieved in the sprint and what they could/couldn’t make, and a sprint retrospective that reviews progress to show to stakeholders and identify lessons and improvements for next sprint

Product backlog, is a list of requirements that the scrum team maintains for a product. It consists of features, bug fixes, non-functional requirements, etc. whatever had be done to successfully deliver a viable product.  
items added to the backlog is commonly written as a story format, the product backlog is ***what*** will be delivered.

## Main Section

The project started by making the **Product Backlog,** The product backlog is when the customer writes down the requirements needed in prioritized order for the system leaving the estimation of hours of work for the team.  
Since we have a REAL customer in this project, we had a list of requirements. We have to of the requirements that the customer wanted us to make, over any of the others.  
The Customer had a list of Requirements it consisted of 7, but we can only make 3 in the short time where 2 of the are the ones he wanted the most

The project consists of 4 requirements: **add Participants, record Score, Real time Results for Parents, Store   
 - Data in DB.**

We selected only a few of those requirements, because of our short timeline. We selected those we thought would be the most important requirements, that is needed around the 2 core requirements our customer came with. The requirements we selected were : Create Database,

We did split the 3 weeks we had into 3 iterations also called Sprints.

### Sprints

First 1st Sprint.

In the first sprint planning meeting, we started to think about what requirements we wanted to do in our 1st sprint, we selected: Sign up, Store Data, create Database,

Second 2nd Sprint.

In the second Spring we talked about what requirements we wanted to implement.   
2nd Sprint we selected: show Score, handle Score points

### Kanban board

## Program Framework

Our group moved on to the structure, how the program should be. A typical structure was used, the 3 layered structure also knows as the MVC structure. See the picture in the appendix how it is reprecents an overview of out MVC.

Need a picture of the MVC and tell how it works with the servlet and so on.. tell what 3 layers we have and what they are named, if they have a handler in each

Tell why we use MVC and what we gain from using an MVC.

## Program Screen Shots

This is some pictures of our program, so that you get a visual effect of what the program looks like running as a reader.

Here you will first be able to see what we have been able to make run and then show or talk to you about what we could not get finish.

### Signing up

### Adding Score

### Watching score Live

# Apendix