



H446/03

Callum Tarttelin, 1164, 51617

Table of Contents

1. Research	1
1.1. Emails	1
1.1.1. The Email	1
1.1.2. Background	1
1.1.3. Handicapping	1
1.1.4. Scoring	2
1.1.5. League Standings	2
1.2. Next stuff	4
2. Planning	4
2.1. What objects, and what do they have	4
2.2. Technologies	5
2.3. Design	6
2.4. Testing	6
2.5. Algorithms	6
3. Development	6
3.1. Initial Setup	6

1. Research

The client currently has a bowling league with an older website. The purpose of this project will be to replace this website with a website easier to use.

1.1. Emails

I received an email from the client about what the project will need to do.

1.1.1. The Email

Welcome to the seemingly easy world of tenpin bowling league management

All of this is currently managed through a number of Excel worksheets linked together, with a load of macros to produce the static web pages and the scoresheets for each week with the updated averages, handicaps and league standings. The web pages are then uploaded to the website each week. It would be great to get to a point where the data entry can be done directly into a [database] on the web server so it can be done from anywhere (and in theory, by anyone). This would help when I am on holiday. Then we can look at webpages that don't need to be uploaded, they would simply get data out of the database. I think it would make sense to meet up so I can show you the mess I work with (which causes headaches at the start of each season when I tweak it to fit – and inevitably break something). That way you'll also be able to ask “why on earth do you do that”, or “you never mentioned that”. Maybe I can show you how I setup a new season, which is when I really find out the bits I need!

1.1.2. Background

The basic structure is that we have a number of teams competing over a number of weeks which make up a season. Each team can have up to 9 bowlers registered and actively bowling. A bowler can switch from one team to another mid-season, however they can only move once during the season. When this happens, their average and handicap move with them. This happens rarely but does happen.

1.1.3. Handicapping

The league runs a handicapping system providing additional points to a bowler based on their current average score. The calculation that we use in Excel is $\text{=INT}(200 - \text{INT}(\text{bowler_average} * 0.8))$. There is a maximum handicap of 80 and a minimum of 0. A bowler with an average of 180 has a handicap of 16 $((200 - 180) * 0.8)$, an average of 180.9 is the same. A bowler with an average of 90 has a handicap of 80 $((200 - 90) * 0.8) = 88$ (more than the maximum) A new bowler joining the league will not have an average, hence no handicap; they will receive a handicap based on their first night's scores. So, if Lucy, in the example below, had bowled those scores on the first night, her handicap would have been 64 based on her average of 120. This then is applied to her scratch scores – she would have won her individual match 6-2 instead of losing 3-5! To keep the handicap current, we use the last 24 scratch game scores to calculate the average for each bowler. Just to be confusing, we do also note the average for all league matches through the season (this is used to determine the “high season average” award – but enough of awards! The “blind” score is simply calculated by adding the bowler's average to their handicap and rounding down... average of 120 = handicap of

64 = blind score of 164 which they would need to beat to win points if the opponent didn't turn up.

1.1.4. Scoring

Each week 3 bowlers per team take part and play 3 games each. In reality, it is more complicated or flexible than this. It could be that each game is played by a different bowler, in the event people get injured; so we need the flexibility for each game to be attributed to a different bowler with a different handicap. Example scoresheet (from the website)

Position	Team	Games	Pins For	Pins Against	HHG Season	HHG All	HHS Season	HHS All	Team Pts	Total Pts
1	Tigers	78	44736	43736	698	698	1903	1903	141	508
2	Strike Force	78	43972	43755	656	656	1802	1802	119	445
3	Hook Line & Sinker	75	42535	42024	662	662	1897	1897	119	432
4	Just Good Friends	78	44855	44592	718	718	1933	1933	108	428
5	Spare Us	78	44057	43603	673	673	1821	1821	111	425
6	Jets	78	44221	43998	650	667	1852	1877	108	422
7	Mid Lane Crisis	78	44005	44463	667	667	1876	1876	111	397
8	Raiders	78	43812	44153	645	645	1797	1797	98	371
9	Razors	78	43557	44657	667	667	1827	1827	84	366
10	Easy Does It	75	41586	42355	684	684	1985	1985	73	304

In the match above Lucy Scott is competing directly against David Henn (first bowler for each team). Their individual match is made up of 4 elements; game 1, game 2, game 3 and the series. I currently record the scratch score (131 for Lucy and 108 for David), their handicap and the bowler name. Once the handicap is added to the scratch score we get the handicapped score, this is what is compared to determine who won the points. In this case Lucy's handicapped score is 189 which beats David's handicapped score of 163, so Lucy is awarded 2 points for game 1. The points are awarded for games 2 and 3 and finally for the series totals. In this instance because the scores are tied (534) they both are awarded 1 point. In theory, we could calculate the handicap "on the fly", however recording the handicap as a static value it allows for the anomalies we see at various times during a season (postponement of matches). Score is each 2 points for a win, 1 point for a draw (equal score). 32 points are available each week: Each bowler 2 points per game (3 games) plus 2 points for the total (series) score with handicap. Scored against their equivalent bowler on the opposite team (based on who bowls 1st/2nd/3rd). So each bowler can score up to 8 points The team total counts as another pseudo-bowler and is scored the same way – total per game and grand-total The handicap score (meaning the bowlers actual or "scratch" score plus their handicap) is always used to work out who won/lost/tied.

1.1.5. League Standings

Mid Lane Crisis										11
HC P	Bowler	Game 1		Game 2		Game 3		Total		Pts
58	Lucy Scott	131	189	124	182	105	163	360	534	3
		2 pts		0 pts		0 pts		1 pts		
36	Keith Biggs	177	213	179	215	168	204	524	632	2
		0 pts		2 pts		0 pts		0 pts		
43	Simon Taylor	159	202	138	181	98	141	395	524	2
		2 pts		0 pts		0 pts		0 pts		
		467	604	441	578	371	508	1279	1690	4
		2 pts		2 pts		0 pts		0 pts		
Spare Us										21
HC P	Bowler	Game 1		Game 2		Game 3		Total		Pts
55	David Henn	108	163	131	186	130	185	369	534	5
		0 pts		2 pts		2 pts		1 pts		
52	Mihir Sampat	180	232	117	169	189	241	486	642	6
		2 pts		0 pts		2 pts		2 pts		
50	Mark Brosnan	134	184	168	218	142	192	444	594	6
		0 pts		2 pts		2 pts		2 pts		
		422	579	416	573	461	618	1299	1770	4
		0 pts		0 pts		2 pts		2 pts		

The league table page shows **HHG = High Handicap Game (the total team score not per bowler)**

***HHS = High Handicap Series (series = sum of the 3 games played any week, again for the team total)**

***Pins for** = total of the scratch scores scored by the team to date

Pins against = total of their oppositions scores each week to date

Team Pts = total points won by the team pseudo bowler each week

Total Pts = all points won by the team

*There is a “Season” and “All” value for each of these – based on the fact we run a cup competition each season. That covers the basics of the scoring and how points are allocated. From all of this, I provide a large number of stats (because I am go through phases of being interested in it.

The Cup The cup competition that runs for a few weeks during the year is also managed this way – but more as an afterthought. The mechanism for determining points can be different to the normal league depending upon how we manage it; normally it’s different when we have an odd number of teams. For each team we see a summary of the score each week and some individual achievements. There is a great deal of detail (which I have put together because it was available)... we can make this “phase 2” Ian’s comments I haven’t addressed elsewhere Some weeks teams will agree to postpone their game. They catch up the missed games later in the season

– This is where the static handicap is useful, with the option to override it. Bowler’s stats are rolled over between seasons (so they continue with the average from last season at the start of a new

season)

- Their last 24 games are used to roll their average to the new season. If they haven't bowled 24 games, then up to 24 games are carried over; if it's less than 6 games then they will start as if new. All achievements are carried over (awards for a 200 game, or 6 consecutive strikes)

There could be a different number of teams some seasons

- If it's an odd number the league make the decision to either bowl against the "blind", or to have a week of no bowling. A team could drop out mid-season and scores must be removed from each weeks results (is that right Phil?)

- It depends, I will check the constitution; if they have bowled against each team just once but a couple of teams twice, we might remove the extra couple of results. We normally encourage them to bowl to a natural break point, but it's not always possible. At the start of the season the system needs to generate a list of matches – which team is playing which other team and on which lanes. The teams need to be put on each lane an equal number of times as far as possible and play each other team twice (maybe 3 times if the number of teams is low enough and there are enough weeks available). If the league has 10 teams assume lanes 1 to 10 are used each week

- We are typically a 10 team league and to be awkward we use lanes 3 to 12 but if we have 10 lane identifiers, we can modify them to be the lanes we actually use.

Some weeks are used for non-league matches (special competitions as teams or individuals tournaments). Some weeks each summer there is no bowling (too maybe people on holiday so we stop). Also bank holidays there is no playing.

- We need a mechanism for recording these scores (we currently use a different area of the spreadsheet to record these, so they are not included within the range used for calculating averages and handicaps but within the range used for awards.

1.2. Next stuff

On top of this existing criteria the current project should have a login system to allow users to add scores for their games. This will need to be confirmed by the other team.

2. Planning

2.1. What objects, and what do they have

```

League{
    teams: [Team, Team ...],
    rota: [Game, Game ...]
    ranking: computed
}

Team{
    name: String
    Image: Image
    players: [Player, Player ...]
    score: Score Object
}

Player{
    name: String
    score: Score Object
}

User{
    ID: 47q047309-47120-97410-298490
    team: Team
    player: Player
    rank: leagueAdmin || teamOwner || scoreAdder
}

Game{
    score: Score Object
    time: yyyy/mm/dd
    venue: Venue
    status: complete || in progress || not started
}

```

2.2. Technologies

I will use springboot and java for the backend. I will use this as java runs well on many platforms. Springboot makes it easy for me to add things into the project. The front end will be written in React JS. I used this as it is very easy to find documentation and sources on how to write it. I will also use material design for styling. The database structure will probably be best in a relational database so a springboot SQL database will be used such as JPA I will use nightwatch for integration testing. This will allow me to automatically use the website, expect behaviour, and take screenshots during. I will use a python script to put all the testing data into the writeup automatically. To allow easy script modification of the writeup it will be written in asciidoctor-pdf, this allows it to be easily exported into a pdf and the source is plain text and can be easily modified. It supports code highlighting etc as well.

2.3. Design

Upon talking to the client it was established that all original features of the site were important. There will be support for multiple types of tournament including elimination and round robin brackets.

A logged in user will be defaulted to a page of their and their teams statistics.

The style is not important other than the website should be mobile compatible.

There should be support for multiple leagues.

To start a new season of a league it should duplicate the previous and delete parts. To move players only a leagueAdmin can move them.

A shortcut should be made to use the API without a frontend to allow for scripts to do a task faster.

2.4. Testing

Testing will be automated by use of unit testing and nightwatch.

Nightwatch will be used in order to test the entire application by it's user interface and report on whether or not it is working as intended. It will also take screenshots of the application to allow a developer to quickly look over these screenshots as opposed to having to navigate the website. This allows for quick testing of all versions without the necessity for user input.

Unit testing will be done to check the functionality of functions. If a function works correctly the test will pass and will then be reported as such. This allows for observation of individual functions to find where errors are occurring.

The data will then be written into the writeup by script in order to have a repeated structure of testing.

2.5. Algorithms

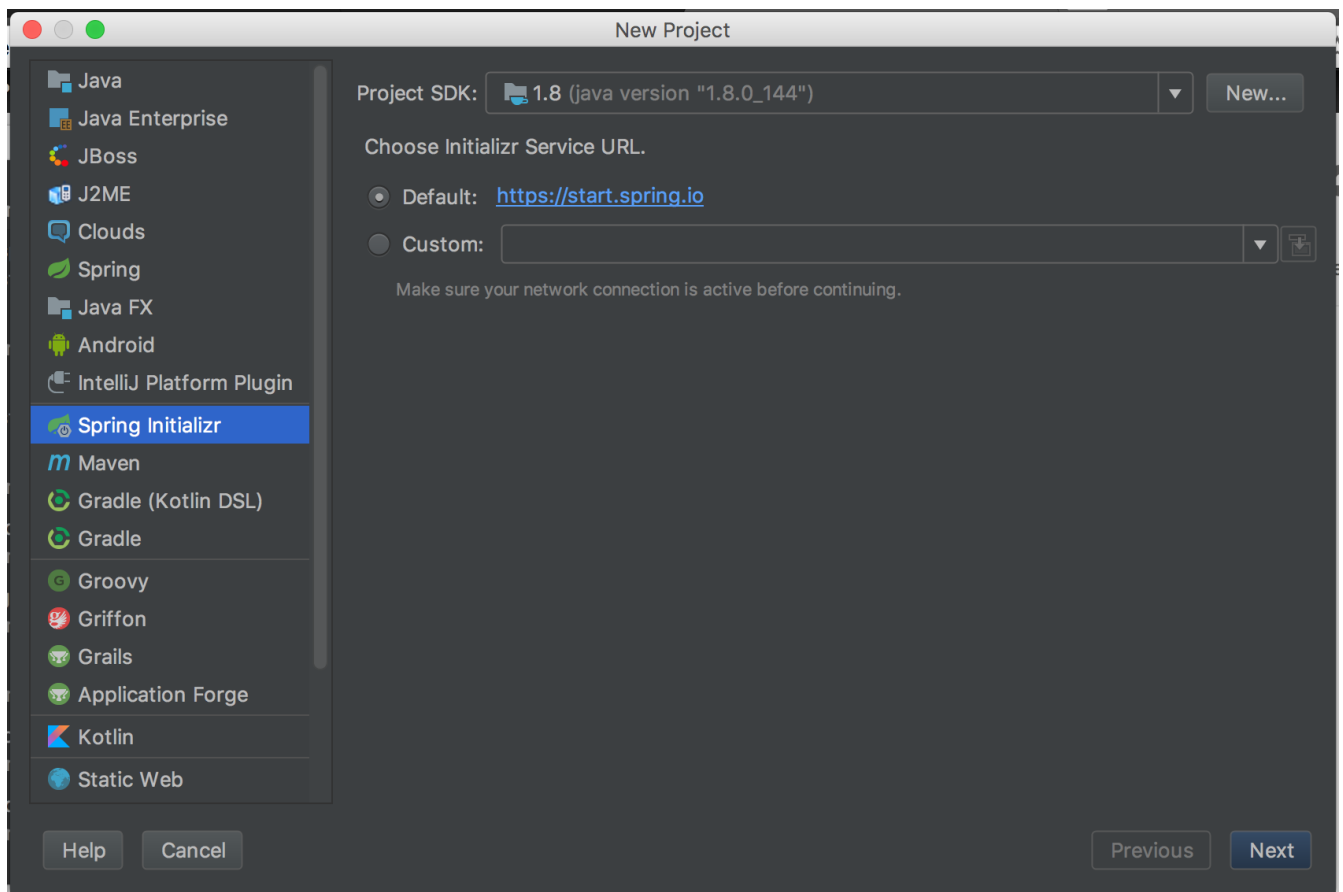
Adding a new season

Copy the old season as a league, change the start and end times, regenerate the rota, allow user to change the rest, Link to old season data.

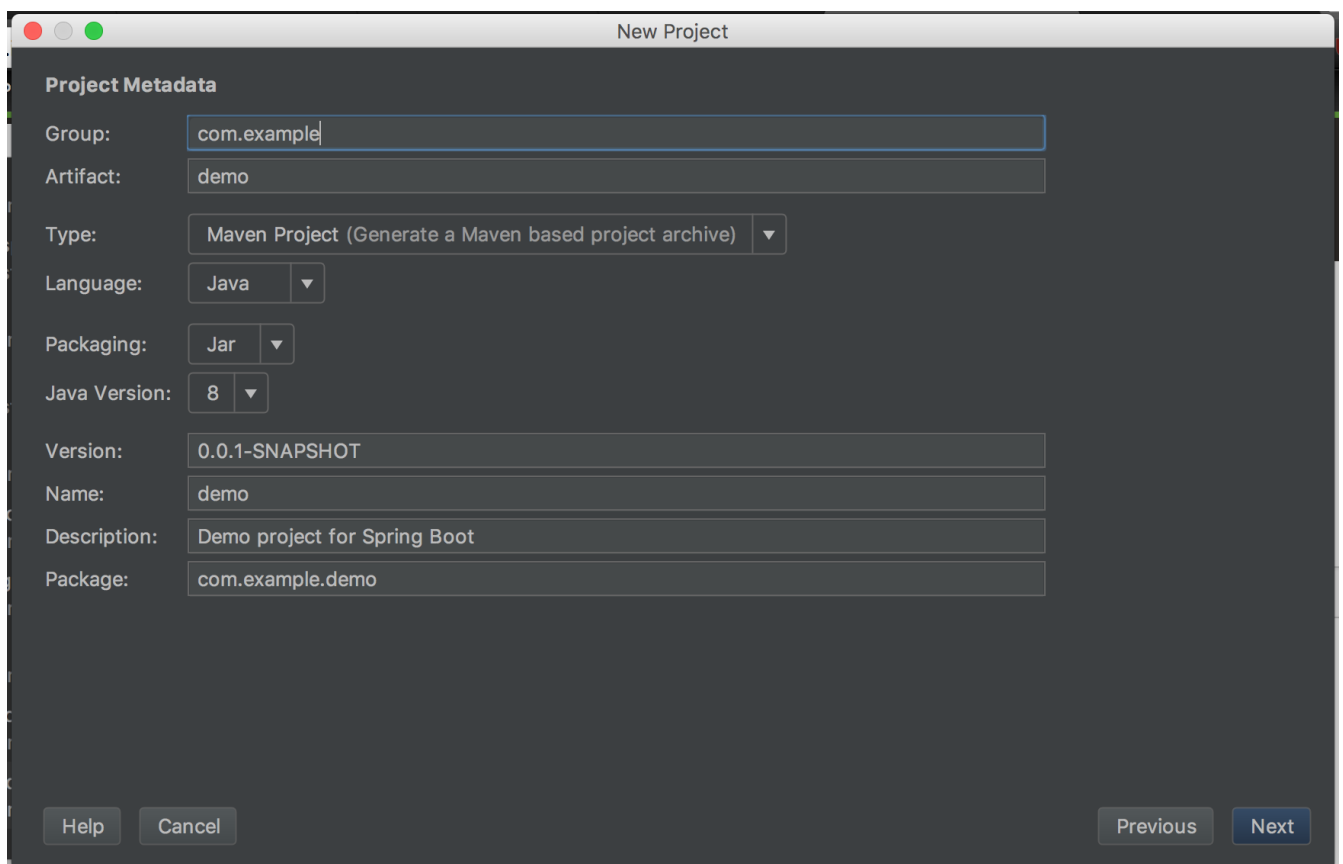
3. Development

3.1. Initial Setup

Using IDEA and springboot I can generate a springboot project with certain libraries pre added.



Step 1: New Project



Step 2: Springboot Project

New Project

Project Metadata

Group:

Artifact:

Type:

Language:

Packaging:

Java Version:

Version:

Name:

Description:

Package:

Buttons: Help, Cancel, Previous, Next

Step 3: Setting up metadata

New Project

Dependencies

Spring Boot

Selected Dependencies

Core

- ☒ Security
- ☐ Aspects
- ☐ Atomikos (JTA)
- ☐ Bitronix (JTA)
- ☐ Narayana (JTA)
- ☐ Cache
- ☐ DevTools
- ☐ Configuration Processor
- ☐ Validation
- ☐ Session
- ☐ Retry
- ☒ Lombok

Core

- Security x
- Lombok x

Web

- Web x

Template Engines

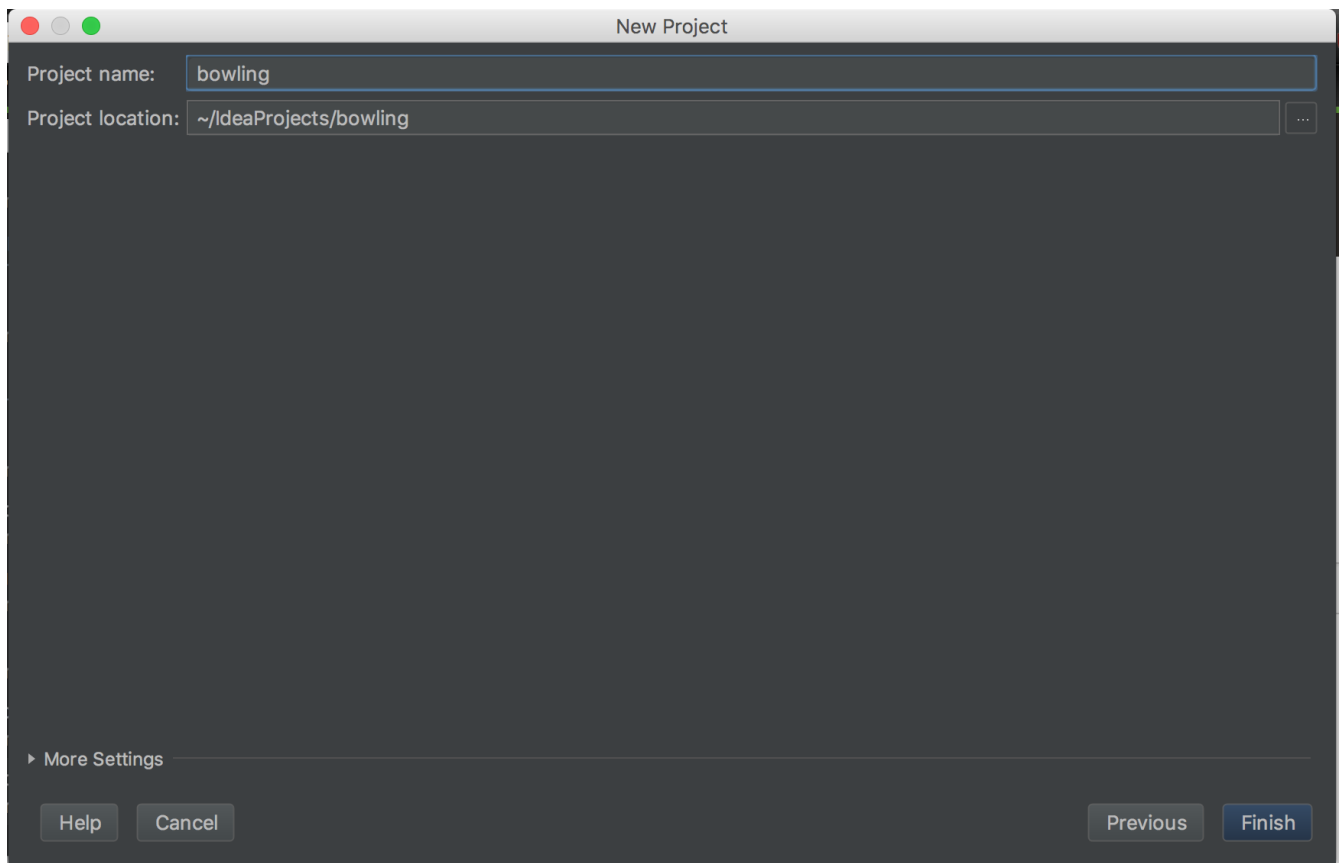
- Freemarker x

SQL

- JPA x
- H2 x

Buttons: Help, Cancel, Previous, Next

Step 4: Choosing dependencies



Step 5: Naming Project