Subsurface
Customizable print formats

Gehad M. Elrobey

gehadelrobey@gmail.com

Table of contents

I. Introduction

Although Subsurface is a mature dive log application, it is missing good customized printing support. This projects aims to add customizable print support to Subsurface so the user can control which information is printed where and using which style, and control over major style elements like font sizes and positioning of the data.

I was a GSoC student for Subsurface last year, and I gained a lot of experience from Subsurface, I hope I will add better contributions to Subsurface this year and also get my hands in more areas in the codebase.
I selected this idea because it fits my skill set well, I love C++ and Qt, I know HTML and CSS and I am very interested to work on this Idea.

II. Requirements

- Customized printing layouts (where to position my data?)
    - Select between pre existing layouts
    - Add special layouts.
- Customized printing styles (how to position my data?).
    - Select colors and fonts in the print dialog.
    - Select page margins.
    - Add special CSS formats in the layout templates.
- Expandability
    - Inherit from existing layouts to create new layouts.
- Choose the data to print (what to print?)
    - Print dive data
    - Print dive table
    - Print dive photos
    - Print dive statistics
        - Print statistics charts/curves

III. Design

Design principles

- Easy for beginners and very customizable for advanced user.
- Minimum navigation between panels.
- Expandability

1. **Custom prints module**
   This is the main part of the project, this module will add the core logic of the printing service.

   Classes:

Printer	This is the main class that holds the rendering logic, the QWebView and QPrinter objects, this class will also lay the dive profile above the rendered HTML page to prevent using svg files as an intermediate stage.
PrintRequest	This class will contain the print type, dives, and print settings, will be used to generalize a print order from FacebookManager or PrintDialog class.

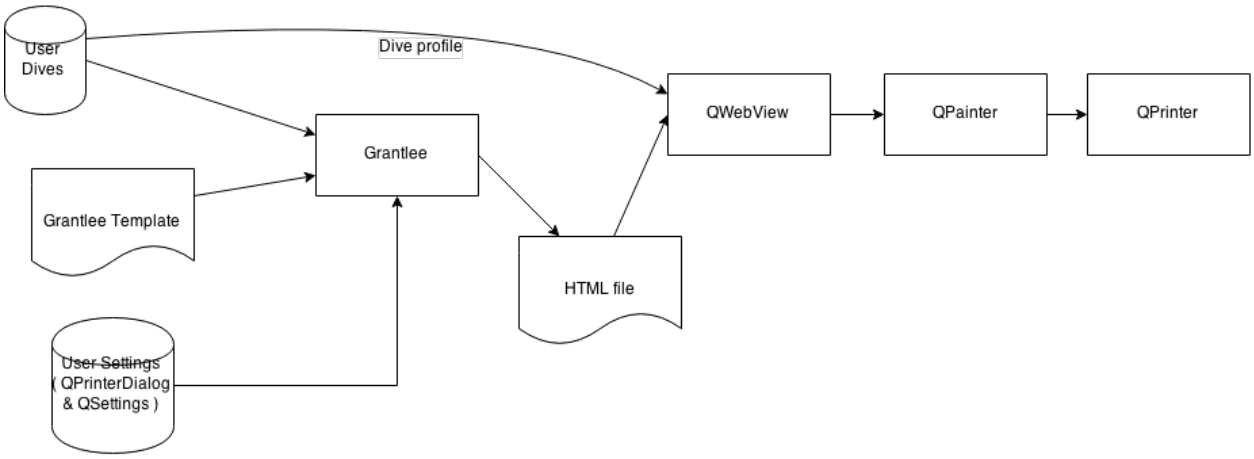
2. **Grantlee backend**
   This part will be mainly dealing with Grantlee to export the HTML file based on the provided template.

   Classes:

TemplateLayout	This class will hold the Templating engine logic.
TemplateEdit	This class will contain the template editing options as well as the logic to customize new templates.

3. **Grantlee templates**
   This are the pre-existing templates that can be used directly.

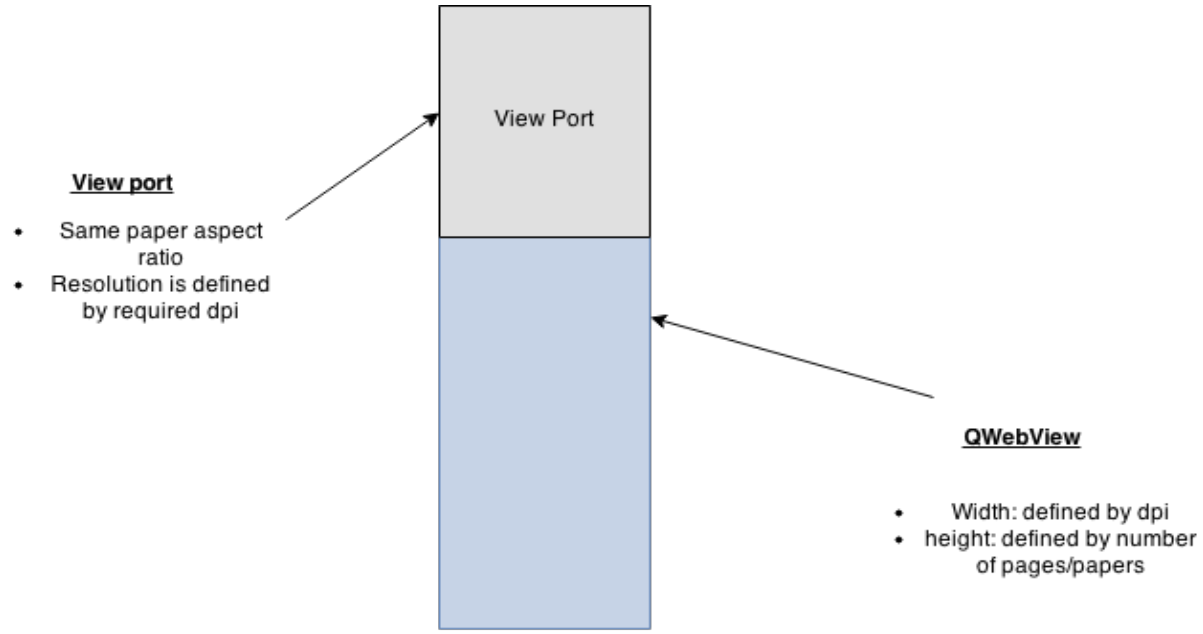
   - 1 dive per page
   - 2 dive per page
   - 4 dives per page
   - flowlayout
   - column flow layout
   - statistics
   - table



Figure(3.1) Printing block diagram

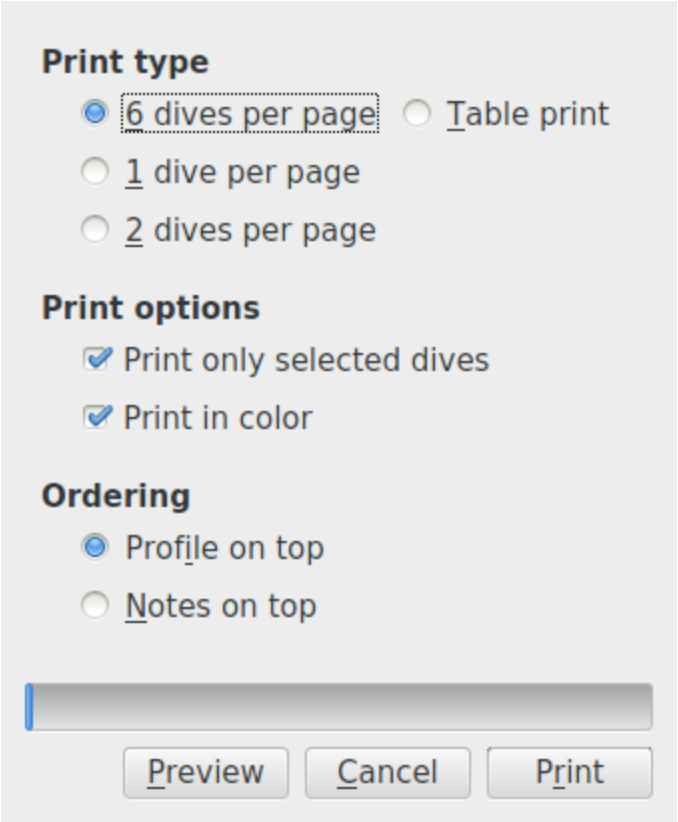
Figure(3.1) shows an overview of the project, Grantlee templating engine will be provided by input from the Template, selected user dives and the print settings from (QPrinterDialog and QSettings), Grantlee will format Html files based on the provided input, The produced temporarily Html file will be rendered on QWebView object, Additionally the dive profiles will be rendered on top of the Html and printed using QPainter.

This will require setting the QWebView content width by determining the page size and the DPI, while the page height will be depending of the content in the html file.
The QPainter viewport width and height will be set by both the page size and the selected DPI. Rendering the QWebView will take place by scrolling the QPainter viewport over the whole content as shown in Figure(3.2)

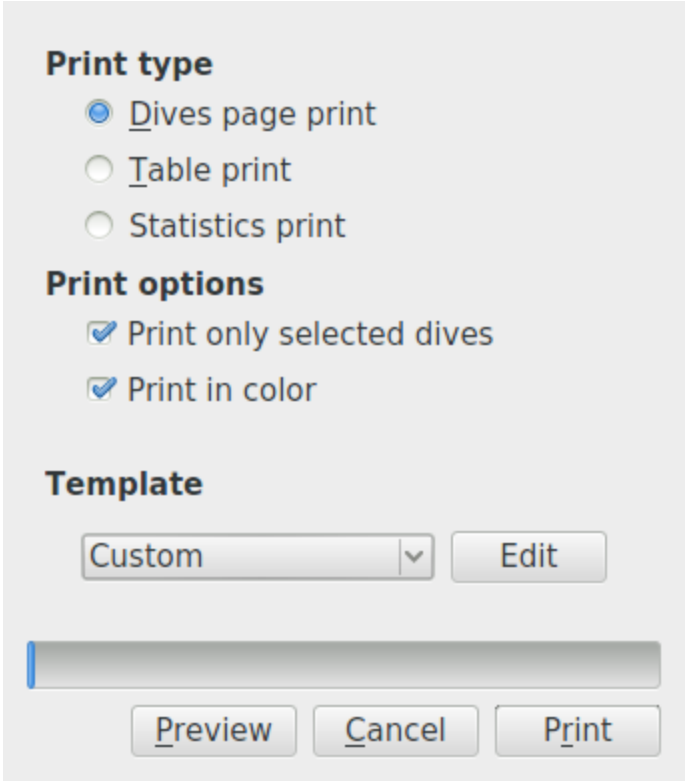


Figure(3.2)

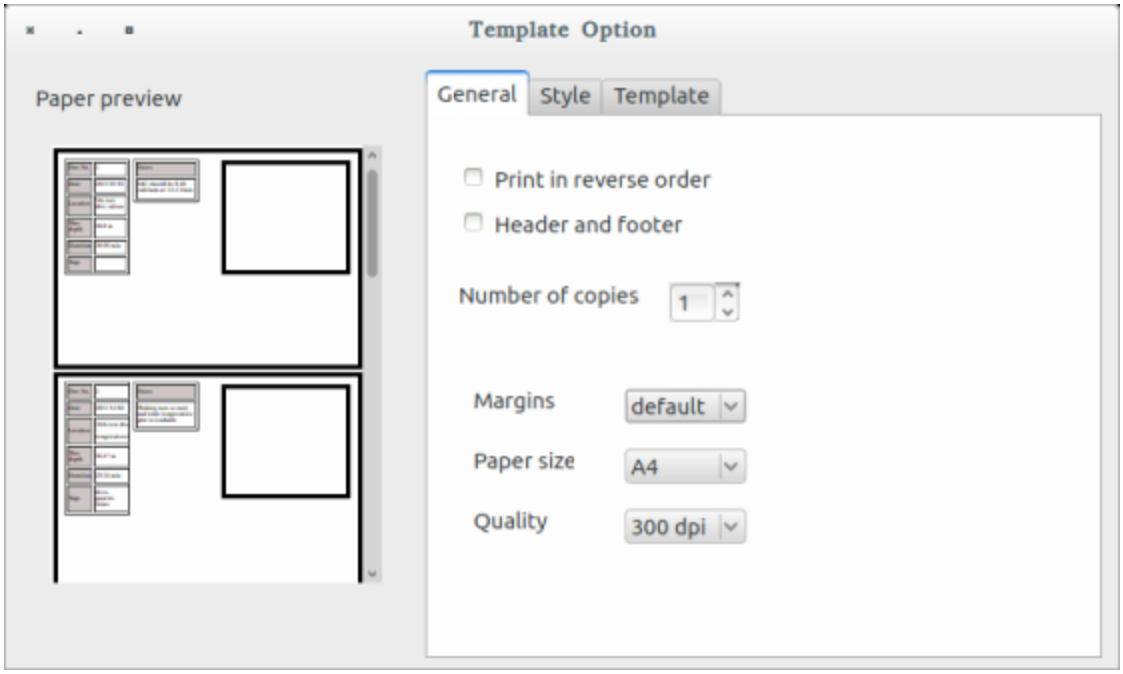
The proposed new dialog figure(3.4) is very similar to the current dialog, the “Ordering section” will be replaced by “Template” section. All existing templates can be chosen from the combobox, also they can be edited by the Edit button.



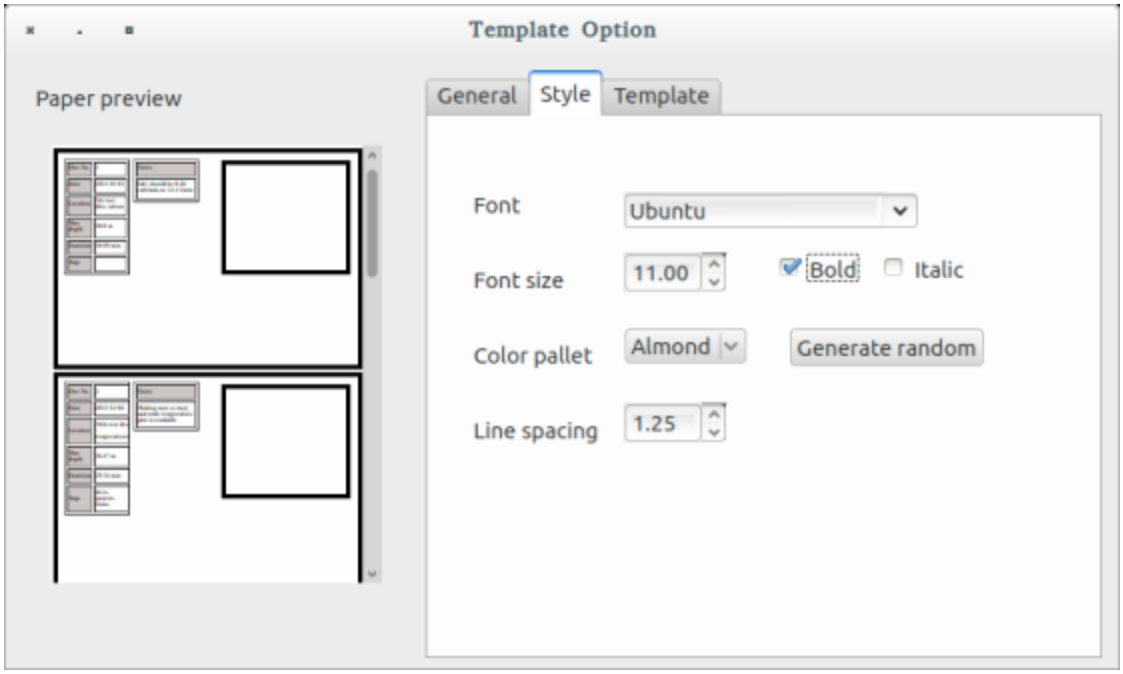
Figure(3.3) - Old print dialog



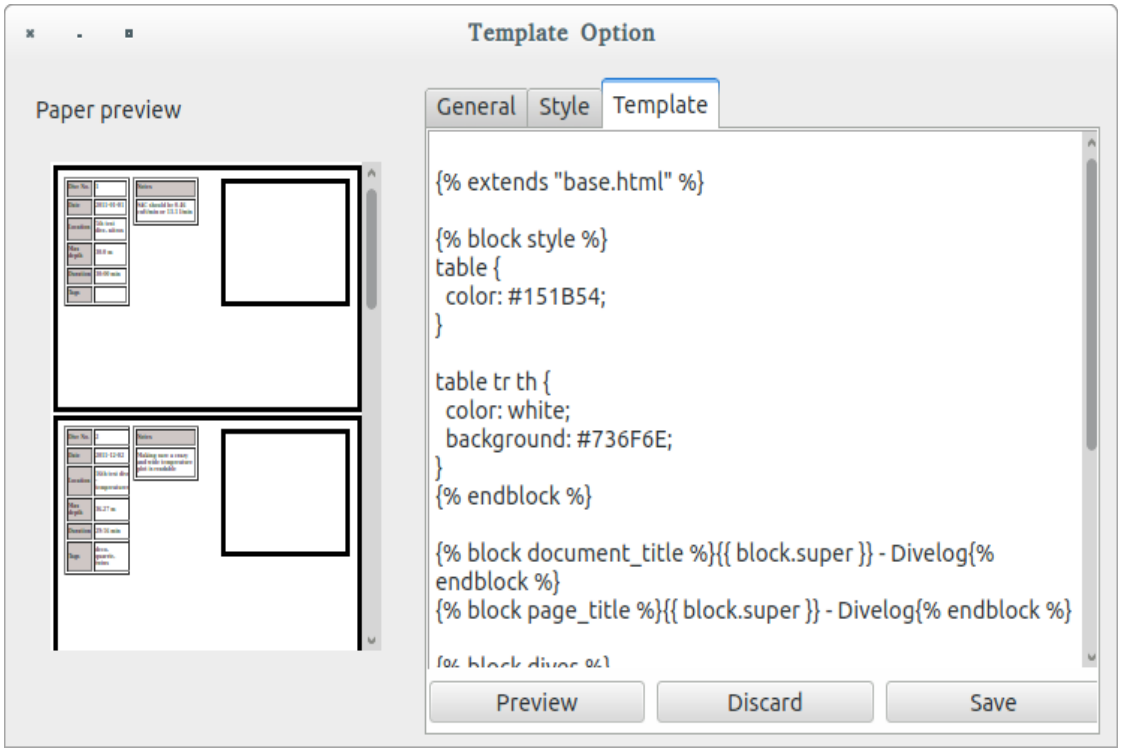
Figure(3.4) - Proposed print dialog



Figure(3.5) - Template options General tab



Figure(3.6) - Template options Style tab



Figure(3.7) - Template options Template tab

The template option window is the main window for template customization, There are three tabs for preferences, The QWebView at the left of the dialog will be used to preview a dynamic version of the user printed files, also this may show a default content if the user has selected no dives.

- General
  user can choose paper size, printing quality, margin size.
- Style
  user can control the font, font size, and colors.
  random color generator can be included.
- Template
  this will add the ability to change the source code of the template, this will provide very advanced customization and the ability to change where and how does the data appear.

1 dive per page

This is a basic 1 dive per page print template, With many dive details are shown

- big dive profile
- dive location, time, temp, etc.
- dive notes
- dive equipments

Figure(3.8)

2 dives per page

This is a 2 dives per page print out, with smaller dive profile in the view, but most important data are attached.

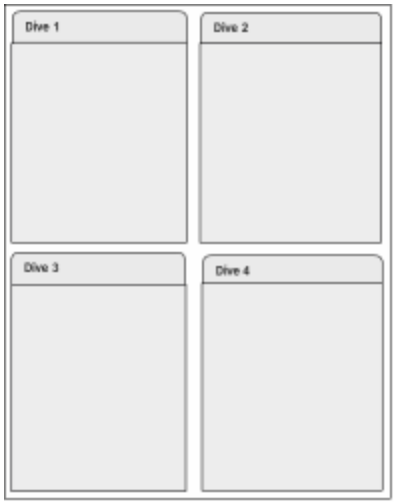
- dive profile
- dive location, time, temp, etc.
- dive notes (can be truncated)

Figure(3.9)

4 dives per page

This is a more compact print out with 4 dives per page, with less space for dive notes.

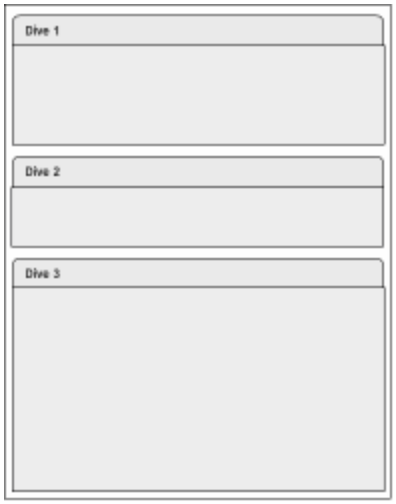
- dive profile
- dive location, time, temp, etc.
- very limited space for dive notes



Figure(3.10)

Flow layout

Very adaptive layout, this should put as many dives in one page. This is specially designed to prevent long dive notes from truncation/overflowing. So the dive height will increase to hold all the dive data.



Figure(3.11)

Column flow layout

A variation of the Flow layout, it divides the page into two columns and will place the dives with dynamic height to prevent truncation just as the Flow layout.

Figure(3.12)

IV. Plan

This project can be describe as four main components, I ll work in an incremental way.

A. Backlog

<table>
<tr><td>Templates</td><td><table>
<tr><td>description</td><td>Develop the grantlee templates</td></tr>
<tr><td>goal</td><td>by the end of this Milestone all the proposed templates must be developed</td></tr>
</table></td></tr>
<tr><td>Grantlee Backend</td><td><table>
<tr><td>description</td><td>Implement Grantlee backend</td></tr>
<tr><td>goal</td><td>by the end of this Milestone HTML files must be generated correctly</td></tr>
</table></td></tr>
<tr><td>Printer Module</td><td><table>
<tr><td>description</td><td>Implement custom printing module</td></tr>
<tr><td>goal</td><td>The produced html files will be rendered correctly and printed</td></tr>
</table></td></tr>
<tr><td>GUI</td><td><table>
<tr><td>description</td><td>Finish the GUI and integration to Subsurface</td></tr>
<tr><td>goal</td><td>by the end of this milestone the editing windows must have been finished and all integration has been finished</td></tr>
</table></td></tr>
</table>

B. Timeline

I am available mostly all the time of the program, I have final exams in the period from 25 may to 9 June, So my time dedicated to the program will be smaller at this period.
The following tables shows my working hours dedicated to the program.

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

May 2015

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

June 2015

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

July 2015

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

August 2015

Availability

- Red: working 3-4 hours/day -Final exams-
- Yellow: working 5-6 hours/day
- Green: working 8-9 hours/day

Task	From	To	Duration
I will increase my experience in Grantlee and review the current printing codebase.	now	Beginning of GSoC	-
Integrate Grantlee and webkit	25 May	28 May	3 days
Implement 1 dive per page template	28 May	2 June	5 days
Implement flow layout template	2 June	6 June	4 days
Implement “Printer” class	6 June	14 June	8 days
Adding dive profiles to the prints	14 June	19 June	5 days
Implementing Grantlee backend to generate dynamic Html files	19 June	26 June	7 days
midterm evaluation (by the midterm the flow layout template and 1 dive per page must be working correctly)	**26 June**	**26 June**	**-**
Integrate the Printer class to printDialog	26 June	27 June	1 day
Implementing the edit template GUI module	27 June	3 July	6 days
Implementing the edit template logic and save QSettings	3 July	7 July	4 days
Implement the remaining templates	7 July	16 July	9 days
Finish grantlee backend	16 July	22 July	6 days
Debugging and testing the custom printing module	22 July	27 July	5 days
Finish integrating all the code and testing (This includes removing the old printing classes)	27 July	4 August	7 days
Refactoring and polishing the code	4 August	8 August	4 days

Documentation and Online tutorial	8 August	11 August	3 days
Supporting Facebook export with the printing backend (**Optional**)	11 August	17 August	6 days
pencils down	**17 August**	**17 August**	**-**

V. Integration

I will Integrate Grantlee and the needed Qt modules in subsurface before I start working, Additional I ll work on a forked branch from subsurface master and rebase regularly with Master. my branch may be forked with subsurface after the midterm since most of the code will be completed by that time.

VI. Bug Fixes

Bug fixes will have the priority over implementing new features, I also added time specially dedicated to debugging before the end of both Milestone 3 and 4.

VII. Documentation

A. User-Manual

I ll document the new printing features in the user manual.

B. Online tutorial

I will write an online tutorial (may be on subsurface-divelog.org) to describe how to create a new template and use it with subsurface printing module from scratch.

VIII. Subsurface Question

A. When did you first hear about Subsurface?
Last year before applying to GSoC 14'

B. Do you have any diving experience? (this is not essential, we have devs that are not divers)
No, I didn't. My intention was to go diving last year after I finished my gsoc project, didn't get the chance anyway hopefully I will dive this year.

C. What attracted you to Subsurface?
Subsurface is an open minded organization with a friendly team of experienced programmers some of them are very popular open source leaders, they welcome contributions from everyone.

D. What other open-source tools do you use?
git, libreoffice, eclipse, openCV, gcc

E. Describe any participation by you in the Subsurface community (e.g. created tutorials, submitted bug reports, asked or answered questions on mailing lists or IRC).
I have submitted some bug reports on trac, I also hangout sometimes on IRC.

F. Describe any contributions you have made to Subsurface development (e.g. bug fixes, translations, packaging, testing).
I have finish the exporting html logbooks idea for gsoc 14' I have also submitted some patches with bug fixes in the QT ui and some other random work.

G. In exactly two sentences, why should we pick YOU?
Because I am very motivated to work on this project and I have the needed skill set. I also want to have the opportunity to be a more involved Subsurface contributor and mentor GSoC projects in the future.

H. What is your name, email address, and irc nickname?
I am Gehad El Robey
email: gehadelrobey@gmail.com
irc: gehadelrobey

I. Describe any plans you have for the summer in addition to GSoC (classes, thesis, job, vacation, etc.).
I don't have any plans for this summer until now, I may travel for one week after the midterm evaluation.

J. What programming projects have you completed?

As I mentioned I have worked on GSoC last year under Subsurface organization, other than that I have worked in customizing Moodle as an adaptive elearning system (Summer internship - Mentor graphics egypt).
I have also completed many programming projects at college using Java, c++ tiny c compiler, equation solver, some java games and some network applications.

K. What are your favorite programming tools (editor, etc.)?

editor: gedit
compiler: gcc
version control: git
build: gnu make

L. List other GSoC projects you are applying to.

I am not applying to any other GSoC projects this year.